



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UNICEUB
FAET- FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUIZ CARLOS BORGES FILHO

ACIONAMENTO REMOTO DE DISPOSITIVOS
ELÉTRICOS UTILIZANDO TECNOLOGIA DTMF E
MICROCONTROLADORES

Brasília – DF
2007



LUIZ CARLOS BORGES FILHO

**ACIONAMENTO REMOTO DE DISPOSITIVOS
ELÉTRICOS UTILIZANDO TECNOLOGIA DTMF E
MICROCONTROLADORES**

Trabalho apresentado à banca
examinadora da Faculdade de
Ciências Exatas e Tecnológicas,
para conclusão do curso de
Engenharia da Computação.
Prof. Orientador: M.Sc. Francisco
Javier de Obaldia.

**Brasília - DF
2008**

LUIZ CARLOS BORGES FILHO

**ACIONAMENTO REMOTO DE DISPOSITIVOS
ELÉTRICOS UTILIZANDO TECNOLOGIA DTMF E
MICROCONTROLADORES**

COMISSÃO EXAMINADORA

Prof. Orientador

[Nome do Examinador]

[Nome do Examinador]

Brasília-DF, dezembro de 2008.

A minha família e aos meus
amigos.

AGRADECIMENTOS

Ao meu pai Luiz Carlos Borges, minha mãe Lúcia Maria Goulart Borges e aos amigos, Wellington Andrade e Franciso Eliezer que me apoiaram nos momentos mais importantes da realização deste projeto.

Aos meus avós e tios que sempre estiveram comigo nessa longa caminhada.

Aos amigos Klécio Vinícius, Winícius Ferraz, Daniel, Clésio Júnior e Adriano Nishiyama que percorreram juntamente comigo este caminho e me ajudaram sempre que precisei.

Ao meu professor orientador Francisco Javier Obaldia pelo apoio e incentivo.

E a todos aqueles que diretamente ou indiretamente me ajudaram realizar este sonho.

“Paciência, amor próprio e
dedicação são as chaves para o
sucesso.”

SUMÁRIO

AGRADECIMENTOS.....	4
RESUMO	8
ABSTRACT	9
LISTA DE TABELAS.....	10
LISTA DE FIGURAS.....	11
LISTA DE ABREVIATURAS	59
CAPÍTULO 1. INTRODUÇÃO	12
1.1 MOTIVAÇÃO	12
1.2 OBJETIVOS.....	13
1.3 METODOLOGIA	13
CAPÍTULO 2. MICROCONTROLADORES E DTMF	15
2.1 DTMF (INTRODUÇÃO)	15
2.2 TECNOLOGIA DTMF - DUAL TONE MULTI-FREQUENCY	16
2.3 APLICAÇÕES.....	19
2.3.1 DTMF EM APLICAÇÕES DE RÁDIOS MÓVEIS.....	19
2.3.2 SISTEMAS DE CONTROLE DISTRIBUÍDOS	20
2.3.3 TRANSMISSÃO DE DADOS USANDO DTMF.....	21
2.4 O RECEPTOR DTMF MT8870	23
2.5 MICROCONTROLADORES FAMÍLIA 8051 (INTRODUÇÃO)	24
2.6 ARQUITETURA BÁSICA DOS MICROCONTROLADORES 8051/8052/80550	27
2.7 ORGANIZAÇÃO DAS MEMÓRIAS	29
2.7.1 MEMÓRIA PARA CÓDIGO (CODE)	29
2.7.2 ESPAÇO PARA MANIPULAÇÃO DE DADOS (DATA)	30
2.7.3 REGISTRADORES PARA FUNÇÕES ESPECIAIS (SPECIAL FUNCTION REGISTER-S.T.R.).....	31
2.7.4 EXTENSÃO PARA MEMÓRIA RAM (IDATA)	32
2.7.5 EXTENSÃO DE MEMÓRIA DE DADOS EXTERNA (XDATA)	32
2.8 O MICROCONTROLADOR AT89S8252 – ATMEL	33
CAPÍTULO 3. PROTÓTIPO	35
3.1 INTRODUÇÃO.....	35
3.2 INTERFACE RECEPTORA DO SINAL DTMF.....	36
3.2.1 PARTE I: DETECÇÃO DE RING E COMUTAÇÃO.	37
3.2.2 PARTE II: RECEPTOR DTMF.....	39

3.3 MICROCONTROLADOR	42
3.4 KIT AT89S8252-ATMEL (FAMÍLIA 8051).....	43
3.5 SOFTWARES DE COMPILAÇÃO E DE GRAVAÇÃO DO MICROCONTROLADOR.	46
3.5.1 O COMPILADOR SDCC.....	46
3.5.2 GRAVA ATMEL	48
3.6 INTERFACE PROJETADA	49
 CAPÍTULO 4. RESULTADOS E A INTERFACE PROJETADA PARA O SISTEMA	 51
4.1 INTRODUÇÃO.....	51
4.2 FUNCIONAMENTO DO PROTÓTIPO.....	51
4.3 PROGRAMAÇÃO DOS EVENTOS PELO USUÁRIO	52
4.4 IMAGENS DA INTERFACE PROJETADA.....	54
 CAPÍTULO 5. CONCLUSÕES E PROJETOS FUTUROS	 56
5.1 CONCLUSÕES.....	56
5.2 PROJETOS FUTUROS	57
5.3 DIFICULDADES ENCONTRADAS	57
 REFERÊNCIAS BIBLIOGRÁFICAS.....	 58
GLOSSÁRIO DE TERMOS	59

FILHO, LUIZ CARLOS BORGES. **Acionamento remoto de dispositivos elétricos utilizando tecnologia DTMF e Microcontroladores**. 2007. 90f. Monografia (Trabalho de Conclusão do Curso de Engenharia de Computação) – Centro Universitário de Brasília - UNICEUB, 2007.

RESUMO

O principal motivo da realização deste projeto está relacionado ao acionamento remoto de dispositivos elétricos.

O grande diferencial deste projeto está baseado na utilização de duas tecnologias: A primeira, tecnologia DTMF (Dual-tone multi-frequency), que é uma tecnologia totalmente consolidada, difundida e bastante utilizada em todo o mundo. A segunda é a utilização de microcontroladores compatíveis com a arquitetura da “família 8051”.

De forma resumida, a tecnologia DTMF é utilizada como um canal de transmissão dos dados. O microcontrolador é utilizado como gerenciador das execuções das rotinas necessárias ao funcionamento do protótipo.

A intenção principal deste projeto é acionar remotamente um dispositivo elétrico utilizando as tecnologias citadas acima. Como demonstração é acionado um conjunto de “LED's” que indica a interpretação dos comandos enviados via DTMF através de um telefone comum e também retorna uma mensagem sonora ao usuário que indica desta forma o sucesso ou falha da ação realizada .

PALAVRAS-CHAVE: Dtmf - Microcontrolador - remoto - acionamento

FILHO, LUIZ CARLOS BORGES. **Drive remote electrical appliances using DTMF technology and Microcontrollers**. 2007. 90f. Monografia (Trabalho de Conclusão do Curso de Engenharia de Computação) – Centro Universitário de Brasília - UNICEUB, 2007.

ABSTRACT

The main reason for the realization of this project is related to the drive to remote electrical devices.

The big difference this project is based on the use of two technologies: The first, technology DTMF (Dual-tone multi-frequency), a technology that is fully consolidated, and fairly widespread use around the world. The second is the use of microcontrollers compatible with the architecture of the "family 8051."

In summary, the DTMF technology will be used as a channel for data transmission. The microcontroller will manage the implementation of the routines necessary for the operation of the prototype.

The main intention of this project is to trigger a device remotely using electric technologies cited above.

KEYWORDS: Dtmf - Microcontroller - remote - drive

LISTA DE TABELAS

TABELA 2.1 – PARES DE FREQUÊNCIAS DTMF.....	17
TABELA 2.2 – REGIÃO “DATA” DA MEMÓRIA.....	31
TABELA 3.1 - FREQUÊNCIAS DTMF	41

LISTA DE FIGURAS

Figura 2.1 - Espectro de multifrequências DTMF.....	17
Figura 2.2 - Controle de sistemas distribuídos utilizando receptores DTMF. [Application Note MSAN-108 MITEL, 2004]	21
Figura 2.3 - Controle residencial utilizando DTMF. [Application Note MSAN- 108 MITEL, 2004].....	22
Figura 2.4 - Arquitetura interna dos microcontroladores da família 8051. [Nicolosi, 2005].....	27
Figura 2.5 - Desenho externo do chip (família 8051 e família 8052). [Nicolosi, 2005]	28
Figura 2.6 - Diagrama em bloco representando os tipos de memória. [Nicolosi e Bronzeri, 2005]	29
Figura 3.1 - Topologia do projeto “ Acionamento remoto de dispositivos elétricos utilizando tecnologia DTMF e Microcontroladores”.....	35
Figura 3.2 – Parte do circuito responsável pela detecção de ring e comutação.....	38
Figura 3.3 - Configuração sugerida pelo fabricante para o receptor DTMF MT8870.	40
Figura 3.4 - Circuito detalhado do receptor DTMF MT8870.....	41
Figura 3.5 - Circuito detalhado do microcontrolador AT89S8253.....	43
Figura 3.6 - Fluxo de dados do Kit de gravação de microcontrolador AT89S8252 – Atmel	45
Figura 3.7 - Kit de gravação de microcontrolador AT89S8252 – Atmel	46
Figura 3.8 - Desenho do circuito da interface (placa) projetada.....	50
Figura 4.1 - Interface projetada (protótipo).....	54

CAPÍTULO 1. INTRODUÇÃO

1.1 MOTIVAÇÃO

O principal motivo da realização deste projeto está relacionado ao acionamento remoto de dispositivos elétricos. Este método já vem sendo utilizado há bastante tempo para as mais diversas funcionalidades, permitindo assim mobilidade e agilidade na interação homem máquina. O grande diferencial deste projeto está baseado na utilização de duas tecnologias: A primeira, tecnologia DTMF (Dual-tone multi-frequency), que é uma tecnologia totalmente consolidada, difundida e bastante utilizada em todo o mundo. A segunda é a utilização de microcontroladores compatíveis com a arquitetura da “família 8051”.

De forma resumida, a tecnologia DTMF foi utilizada como canal de transmissão dos dados. Comandos são enviados ao dispositivo e o microcontrolador serve como interpretador destes comandos que foram enviados. DTMF é a principal tecnologia utilizada em telefonia fixa e um canal de transmissão de dados que pode alcançar praticamente todos os pontos do planeta que utilizem este mecanismo.

A intenção principal deste projeto é acionar remotamente um dispositivo elétrico utilizando as tecnologias citadas acima. Como demonstração ocorre o acionamento de um conjunto de “LED's” que indicará a interpretação dos comandos enviados via DTMF através de um telefone comum e também retornará uma mensagem sonora ao usuário indicando que ação foi realizada com sucesso ou falha.

1.2 OBJETIVOS

O objetivo principal deste projeto é utilizar as tecnologias citadas em uma implementação de sistema que permita realizar a programação remota (acionamento) em horários preestabelecidos, de dispositivos elétricos. Os comandos serão enviados ao microcontrolador através de um telefone comum. Como demonstração utiliza-se uma seqüência de “LED's” que representa os dispositivos a serem acionados, indicando assim se o dispositivo foi acionado ou desligado corretamente. O usuário também recebe uma mensagem sonora através do próprio telefone informando se o comando enviado foi realizado com sucesso ou não.

1.3 METODOLOGIA

O trabalho está organizado em cinco capítulos. Os dois primeiros capítulos fazem a apresentação do tema do projeto, fornecem o embasamento teórico do trabalho e a tecnologia utilizada no projeto. Nos capítulos seguintes são analisadas as abordagens e técnicas discutidas durante os capítulos iniciais e o último capítulo do trabalho traz as conclusões e os estudos futuros. A organização detalhada está descrita da seguinte forma:

-Capítulo 2: Refere-se a todo o material teórico utilizado na construção do projeto. Abordaremos a situação atual das tecnologias empregadas, forma de transmissão das informações existente e como é usado hoje o DTMF.

-Capítulo 3: Está relacionado à topologia do projeto. O objetivo aqui é enfatizar as ferramentas que serão utilizadas, a definição do hardware mais adequado para a construção do protótipo, o canal de comunicação que será

utilizado para a transmissão de dados e os demais recursos necessários para a implementação do projeto.

-Capítulo 4: Será dedicado à demonstração dos resultados obtidos. Telas contendo as interfaces utilizadas, demonstrativos da comunicação e transferência de informações/leituras do sistema e demonstração da aplicação serão os principais pontos abordados nesta etapa do trabalho.

-Capítulo 5: Fica definido para novas sugestões de projetos, conclusões e análise dos resultados.

CAPÍTULO 2. MICROCONTOLADORES E DTMF

2.1 DTMF (INTRODUÇÃO)

Neste capítulo são apresentadas as principais tecnologias utilizadas no projeto a ser implementado, a forma de transmissão das informações e como é empregada a utilização de microcontroladores, especificamente da família 8051 ou compatíveis com a mesma. Também é abordado o desenvolvimento da sinalização DTMF, como está sendo utilizada atualmente e suas diversas aplicações. Este capítulo serve para descrever o porquê da escolha destas tecnologias para a solução proposta.

A sinalização telefônica é um mecanismo criado para prover às centrais telefônicas as informações necessárias para que seja possível estabelecer uma ligação. Essas informações têm como conteúdo os dados dos terminais que desejam estabelecer uma comunicação, o estado que se encontra o sistema, boletins e alertas referentes aos procedimentos que estão sendo realizados. A importância dos mecanismos de sinalização acompanha a evolução dos meios de comunicação e ganha força com as melhorias feitas para proporcionar aos sistemas telefônicos certa independência no funcionamento e manutenção dos serviços prestados. [FERRARI, 2005]

Nos primeiros telefones a discagem era feita através de um disco que gerava uma seqüência de pulsos na linha telefônica ("discagem decádica" ou "discagem usando sinalização decádica"). Ao se ocupar a linha, o "loop" (laço) era fechado e ao se efetuar a discagem, ocorriam aberturas periódicas deste "loop", tantas vezes quanto o número discado: para a discagem do 1, uma abertura, para a discagem do 2, duas aberturas, e assim sucessivamente até o 0 (zero) que, na verdade, significava 10 aberturas [Application Note MSAN-108 MITEL, 2004]

Com o advento dos telefones com teclado, das centrais telefônicas mais modernas e com a disseminação dos filtros (primeiro os analógicos e depois os digitais), passou-se a utilizar a sinalização multifrequencial, ou seja, uma combinação de tons. Os “DTMFs” vulgarmente conhecidos em inglês por “Touch-Tone”, para realizar discagem [Application Note MSAN-108 MITEL, 2004]

Aproximadamente 25 anos atrás a melhoria do método de transferência de informações (dados) através da rede de telefonia tornou-se extremamente necessária. O método tradicional, sinalização por discagem de pulso, além de demonstrar-se lento era caracterizado pelo excesso de distorções durante as ligações de longa duração. Um esquema de sinalização foi desenvolvido utilizando tons da frequência de voz e implementado como uma alternativa de grande confiabilidade quando comparada a discagem de pulso. Este esquema ficou conhecido como DTMF (Dual Tone Multi -Frequency), ou como já citado anteriormente, “Touch-Tone™”, ou simplesmente, discagem de tom [Application Note MSAN-108 MITEL, 2004]

2.2 TECNOLOGIA DTMF - DUAL TONE MULTI-FREQUENCY

O conceito de um sinal que utiliza a tecnologia DTMF é caracterizado pela soma de dois tons. O primeiro tom deve pertencer a um grupo de frequências baixas que tem uma faixa que varia entre (697-941Hz). O segundo pertence ao grupo de frequências altas definida entre a faixa de (1209-1633Hz). Cada grupo possui quatro tons de frequências individuais. As frequências foram escolhidas com cuidado, de tal forma que não estão relacionadas harmonicamente e que seus produtos de intermodulação resultam num impedimento mínimo à sinalização. Observar a figura 2.1 que mostra os espectros de frequências. [Application Note MSAN-108 MITEL, 2004]

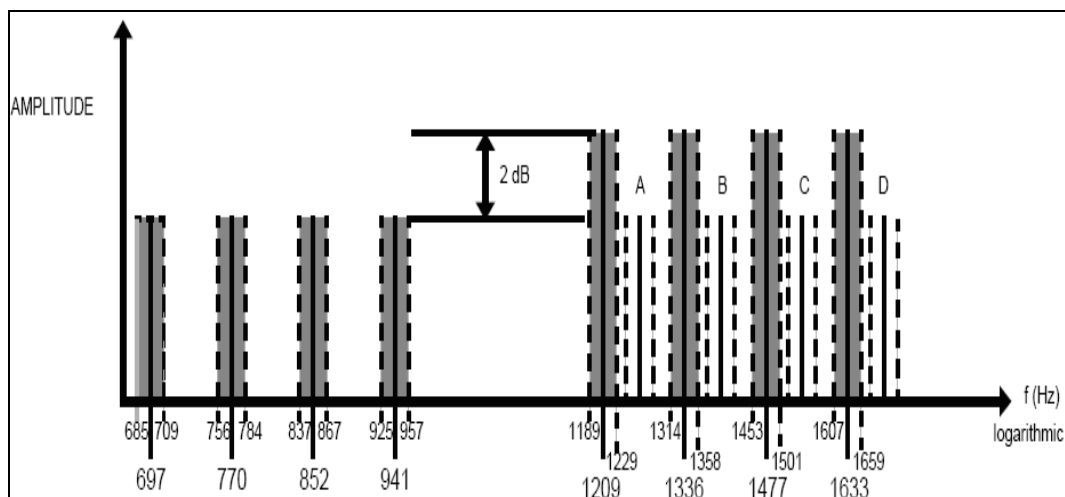


Figura 2.1 - Espectro de multifrequências DTMF

Este esquema permite 16 (dezesseis) combinações originais. Dez destes códigos representam os números de zero a nove e os seis restantes (*, #, A, B, C, D) são reservados para sinalização especial. A maioria dos telefones possuem 10 (dez) teclas numéricas mais o asterisco (*) e o cerquilha (#). As teclas são arranjadas em uma matriz. Cada tecla é determinada por um tom que está dentro do grupo das baixas frequências, representadas pelas linhas da matriz, e por um tom do grupo das frequências elevadas, representadas pelas colunas da matriz, respectivamente: [Application Note MSAN-108 MITEL, 2004]

Tabela 2.1 – Pares de frequências DTMF

Hz	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Na tabela 2.1 são mostradas as frequências "altas" na linha superior e as baixas na coluna mais à direita. No centro os números do teclado. Nos teclados dos telefones são mostrados apenas os números de 1 até 0 e os caracteres "*" e "#". A frequência de 1633 hertz (e conseqüentemente os algarismos "A", "B", "C" e "D") é utilizada apenas internamente entre

equipamentos de teste e medida. O esquema de codificação DTMF assegura que cada sinal contém um e somente um componente de cada um dos grupos de frequências elevadas e baixas.[Application Note MSAN-108 MITEL, 2004]

Esta característica simplifica significativamente a decodificação porque o sinal composto de DTMF pode ser separado com filtros passa-banda. Com isto cada componente de frequência do sinal pode ser analisada individualmente. Com esta propriedade a codificação DTMF provou fornecer um esquema de sinalização flexível, de grande confiabilidade, servindo de motivação para o desenvolvimento de novos projetos de decodificadores. [Application Note MSAN-108 MITEL, 2004]

Os primeiros decodificadores foram receptores DTMF e utilizaram bancos de filtros passa-banda que além de difícil implementação possuíam um custo elevado. Isto restringiu a utilização desta tecnologia geralmente às centrais telefônicas. O decodificador (receptor) da primeira geração usou tipicamente filtros LC, filtros ativos e técnicas de bloqueio de fases para receber e decodificar tons de DTMF.[Application Note MSAN-108 MITEL, 2004]

As funções iniciais eram decodificar o número do telefone e restringir a chamada. A introdução de técnicas digitais de MOS/LSI permitiram o desenvolvimento da segunda geração de receptores de tons DTMF. Estes dispositivos foram utilizados para digitalizar a decodificação dos dois tons discretos que resultavam da decomposição do sinal inicial. Dois filtros passa-banda analógicos foram utilizados para executarem a decomposição do sinal. [Application Note MSAN-108 MITEL, 2004]

Os receptores totalmente self-contained (reservado) descreveram o começo de dispositivos da terceira geração. Tipicamente, usaram também filtros ativos analógicos bandsplit no sinal composto e os dispositivos digitais do MOS para decodificar os tons. O desenvolvimento da implementação do silicone em conjunto com os filtros que utilizam capacitores, marcaram o

nascimento da quarta e também atual geração da tecnologia dos receptores DTMF. Inicialmente os filtros passa-banda de um único chip foram combinados com os decodificadores atualmente disponíveis. Com isto, foi possível o desenvolvimento de um receptor com dois chips. [Application Note MSAN-108 MITEL, 2004]

Avanços futuros permitiram a integração de ambas as funções em um único chip permitindo que os receptores de DTMF possuam dimensões reduzidas e baixo custo. A segunda e terceira geração demonstra uma tendência de deslocar a complexidade dos circuitos analógicos para os circuitos digitais do LSI a fim reduzir a complexidade de filtros analógicos e de seus problemas inerentes. Agora que os filtros podem ser implementados utilizando silicone, a distribuição da complexidade torna-se mais uma função do desempenho e das propriedades físicas do material, neste caso o silicone. [Application Note MSAN-108 MITEL, 2004]

2.3 APLICAÇÕES

A confiabilidade demonstrada na sinalização DTMF possibilitou a implementação de diversas aplicações. Com o desenvolvimento de um receptor DTMF completo em apenas um chip e algumas outras vantagens a relação custo benefício torna-se bastante satisfatória.

2.3.1 DTMF EM APLICAÇÕES DE RÁDIOS MÓVEIS

Muitos dos nossos equipamentos de rádios, especialmente os "walky-talkies", dispõem de teclado DTMF e podem enviar os códigos a outras estações de rádio, sistemas de alarmes, telemetria, telecontrole de

repetidores e outros.

O DTMF possui um papel importante nos sistemas de comunicações distribuídos, tais como o rádio móvel. Possui também excelente imunidade a ruídos e disponibilidade de métodos de implementação altamente integrados. [Application Note MSAN-108 MITEL, 2004]

Diversas redes e serviços de emergência médica utilizam atualmente sinais de DTMF para controlar rádio repetidores. As funções são, tipicamente, identificação entre transmissores e receptores móveis, seleção das ligações apropriadas do repetidor, seleção de frequências do repetidor e leitura de status do repetidor e ligações automáticas para telefones. Se disponível em um sistema deste tipo, o som de uma ligação de longa distância (microondas, satélites, etc) pode ser comutado através de comandos de um teclado DTMF operado por um usuário, para o local repetidor. [Application Note MSAN-108 MITEL, 2004]

2.3.2 SISTEMAS DE CONTROLE DISTRIBUÍDOS

Há muitas outras aplicações que caem também na chamada classe de controle de comunicações distribuídas. Isto é, diversos dispositivos que são controlados através de um meio de comunicação comum, quer seja o RF, fio de cobre ou fibras ópticas, etc. [Application Note MSAN-108 MITEL, 2004]

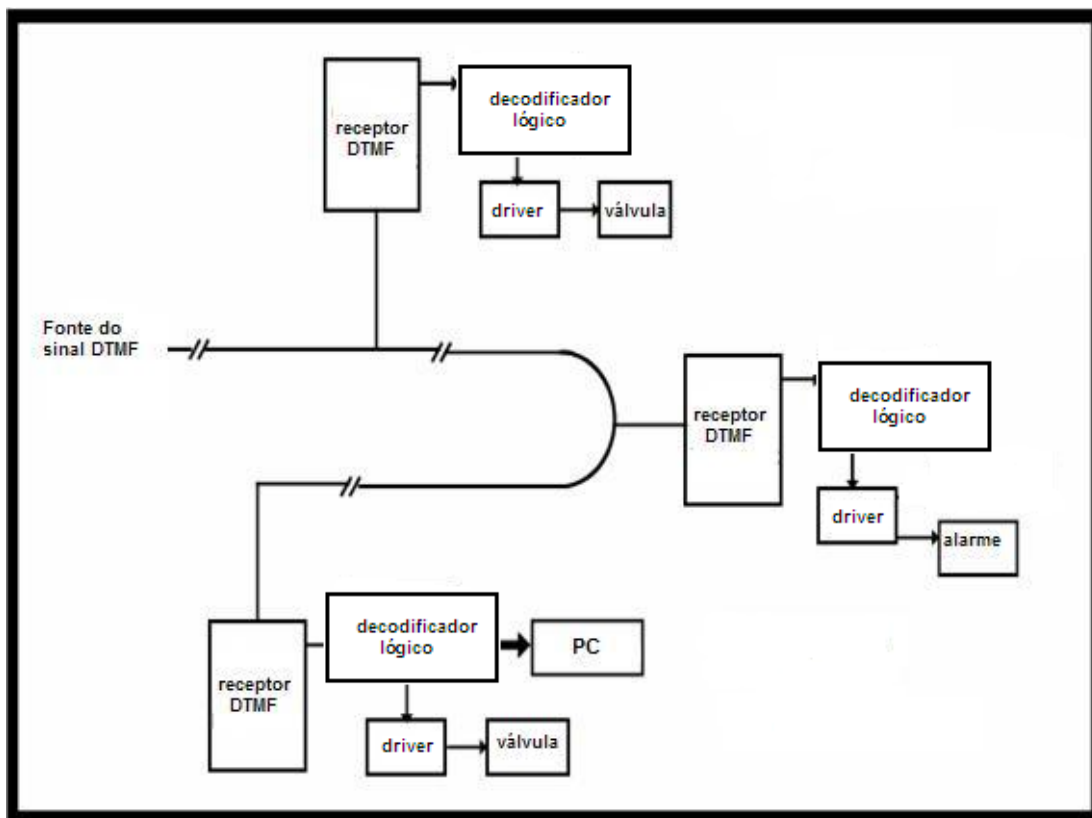


Figura 2.2 - Controle de sistemas distribuídos utilizando receptores DTMF. [Application Note MSAN-108 MITEL, 2004]

De acordo com o modelo descrito na figura 2.2 cada receptor DTMF monitora a fonte do sinal até o seu ID específico, que depois transfere dados a sua lógica de controle funcional. Com uma lógica simples um circuito pode ser planejado para reconhecer uma seqüência de código programado para DTMF. [Application Note MSAN-108 MITEL, 2004]

2.3.3 TRANSMISSÃO DE DADOS USANDO DTMF

Há uma disposição vasta de aplicações potenciais para DTMF que utiliza a rede de telefonia existente. Considerando a existência de pontos que podem realizar a leitura de informações recebidas, o usuário pode enviar dados que serão executados remotamente em outros locais. As aplicações potenciais

incluem: controle remoto (residencial), entrada remota de dados através de teclado do telefone, serviço de consulta automaticamente, como exemplo, operadoras de cartões de crédito. Observe a figura 2.3: [Application Note MSAN-108 MITEL, 2004]

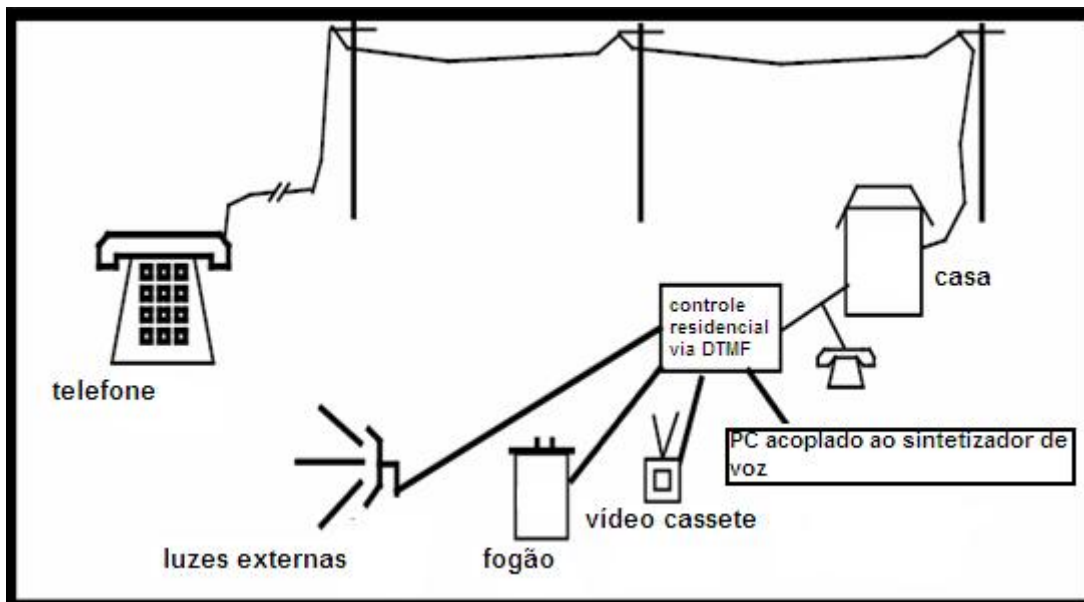


Figura 2.3 - Controle residencial utilizando DTMF. [Application Note MSAN-108 MITEL, 2004]

Um sistema de controle remoto de casa utilizando a sinalização DTMF pode permitir uma variedade de conveniências como: Controle de ligar/desligar remotamente dispositivos elétricos tais como uma TV, uma lâmpada externa, um aquecedor, etc. Um solenóide eletromecânico a válvula operado remotamente poderia acionar o sistema de irrigação de um jardim. Os sistemas da segurança podem ser controlados a distância e um microfone poder ser utilizado para a monitoração remota de áudio. Podemos tornar um PC num grande centro de mensagens através da seguinte forma: As mensagens são enviadas através do teclado do telefone. O PC responde com mensagens de voz geradas através de um sintetizador. As mensagens podem ser exibidas através do monitor ou podem ser ouvidas através de algum auto-falante. [Application Note MSAN-108 MITEL, 2004]

O DTMF proporciona um grande número de aplicações e devido aos avanços tecnológicos sua utilização está se tornando extremamente difundida. Oferece alta confiabilidade, baixo custo e grande eficácia. A

construção de um receptor completo em apenas um único chip, a relação custo-benefício e outras vantagens, permitiram a difusão desta tecnologia e a produção em larga escala.

A sinalização de DTMF foi projetada originalmente para melhorar a qualidade de voz nas linhas telefônicas, mas, devido as diversas qualidades, esta técnica de sinalização foi aplicada a uma variedade de sistemas de controle e de transmissões de dados. São relativamente importantes dois aspectos nesta tecnologia: qualidade no canal de comunicação de voz e uma interface apropriada. As aplicações são limitadas somente a nossa imaginação. [Application Note MSAN-108 MITEL, 2004]

2.4 O RECEPTOR DTMF MT8870

O MT8870D é um completo receptor DTMF que integra um filtro de banda e o decodificador digital de funções. A seção de filtro utiliza a técnica de capacitores comutados para grupos de filtros altos e baixos. O decodificador utiliza a técnica de contagem digital para detectar e decodificar todos os 16 pares de tons DTMF em um código de 4 bits.

Possui um tamanho consideravelmente pequeno, baixo consumo de energia e alto desempenho. A sua arquitetura consiste de uma seção de filtro , que separa os grupos de tons altos e baixos, seguido por uma seção de contagem digital que verifica a frequência e a duração dos tons recebidos antes de passar o correspondente código para o barramento de saída.

Este componente é utilizado para diversas aplicações. Dentre as mais comuns podemos citar:

- rádios móveis.
- sistemas de cartão de crédito.
- controle remoto.

- computadores pessoais.
- secretária eletrônica.

As características apresentadas por este receptor atendem perfeitamente às necessidades do projeto. Baixo custo, pequeno consumo de energia e o desempenho bastante satisfatório foram os principais pontos que determinaram a utilização deste componente no protótipo desenvolvido.

2.5 MICROCONTROLADORES FAMÍLIA 8051 (INTRODUÇÃO)

No início da década de 80, a INTEL, que já tinha criado o microcontrolador 8048, lançou uma nova família, herdeira do 8048, intitulada 8051, de "8 Bits". Inicialmente ela lançou o "8051" com ROM interna programável de fábrica, o "8751" com EPROM interna programável pelo usuário e o "8031" com necessidade de ter "chips" de EPROM externa. Após alguns anos, a família se expandiu para o "8052", que tem um timer a mais que o 8051 e suas variações: com EPROM (8752) e sem ROM/EPROM (8032), e uma versão especial do 8052, chamado 8052 - Basic, que tem dentro de sua ROM um interpretador Basic que permite programá-lo também em linguagem Basic. Esta é a família chamada MCS-51. Desde então, o 8051 tem sido um dos mais populares microcontroladores, possibilitando ter uma vasta família no mercado, sendo hoje produzido por mais de 30 fabricantes, com mais de 600 variações de chips. [Nicolosi e Bronzeri,2005]

As diferentes versões de chips podem apresentar conversores analógico-digital (A/D), variação com relação à quantidade de memória (RAM / ROM internas), modulação PWM e memória Flash (que possibilita a reprogramação mediante um sinal elétrico), conversores analógico-digital (A/D) e digital-analógico (D/A), comunicação SPI (*SPI - Serial Program Interface*), comunicação serial I²C (*Acces Bus*), tecnologia de interconexão para redes locais (*Ethernet*), CAN (*Control Area Network*), memórias EEPROM que podem ser utilizadas pelo programa em tempo real, e clock de

2 a 100MHz. Tudo em apenas um Chip. [Nicolosi e Bronzeri,2005]

Hoje em dia, também encontramos microcontroladores derivados da família 8051 que trabalham com 16 bits, com alta performance, sendo aplicados em processamento digital de sinais e controle de sistemas em tempo real. [Nicolosi e Bronzeri,2005]

Alguns dos microcontroladores de 16 bits possuem grande quantidade de memória, maior número de canais de conversores analógico-digitais, grande número de “ports”, “ports” seriais, alta velocidade aritmética e lógica de operação, e um poderoso set de instruções com capacidade de processamento de sinais. [Nicolosi e Bronzeri,2005]

Basicamente, o 80C51 (80“C”51 - Tecnologia CMOS), observe o “C” no código ao lado, que é de tecnologia “CMOS” a qual possui baixo consumo de energia, fabricado pela Intel, é considerado o hardware básico da família de microcontroladores 8051. Porém, com o avanço tecnológico na elaboração de componentes semicondutores e a grande procura por esses componentes, existe hoje um número expressivo de fabricantes no mercado, sendo a variedade de modelos apenas consequência da quantidade de fabricantes. Dentre eles temos: Mitsubishi, Siemens, Philips, Intel, Toshiba, Atmel, National, Texas, Sharp e outros. São mais de 600 variações da família básica. [Nicolosi e Bronzeri,2005]

Alguns representantes da família Atmel de microcontroladores de 8 e 16 bits Baseiam-se no microcontrolador C251, uma versão mais completa do 80C51. Outros representantes, como o AT89C51CC03, possuem uma interface CAN (Control Area Network), possibilitando comunicações a uma distância de aproximadamente 40 metros, com alta velocidade. Além dessa interface, esse microcontrolador é provido da capacidade de reprogramação sem a necessidade de retirarmos o chip do circuito, o que é feito por software. Os pacotes trazidos pelo sistema CAN são traduzidos em código binário a fim de reprogramarmos a memória Flash. Outros microcontroladores apresentam um sistema similar UART (Universal

Assynchronous Receiver Transmitter), porém a distância para comunicação é menor. [Nicolosi e Bronzeri,2005]

Além da UART, existe uma outra forma de comunicação serial denominada de "ACCESS BUS", também conhecida como I²C. Ele foi criado pela Philips e tem sido adotado como padrão. Possui também grande importância no cenário industrial. Com esse protocolo é possível transferir dados, de forma bidirecional, entre o sistema central (mestre) e um sistema periférico (escravo), ou para outros sistemas mestre. Por ter sua interface dentro do chip, não ocupa espaço na placa, além de possibilitar conexão com vários dispositivos ao mesmo tempo por meio de apenas dois pinos físicos. [Nicolosi e Bronzeri,2005]

Tipicamente o 8051 é um microcontrolador que trabalha com palavras de 8 bits, possui alta performance e baixo custo. Um representante básico de família, contém internamente:

- 64 Kbytes de memória para DATA (RAM) e CODE (ROM).
- 256x8 bytes de memória RAM, dividida entre área de uso geral e registradores especiais.
- Dois timers/counters de 16 bits.
- Uma porta serial programável (UART).
- Interface para memória externa com capacidade de 64 Kbytes de endereçamento externo para ROM e 64 Kbytes de endereçamento externo para RAM.
- Quatro portas de I/O.
- Seis possibilidades de interrupções com dois grupos de prioridades. [Nicolosi e Bronzeri,2005]

2.6 ARQUITETURA BÁSICA DOS MICROCONTROLADORES 8051/8052/80550

Observe a figura 2.4:

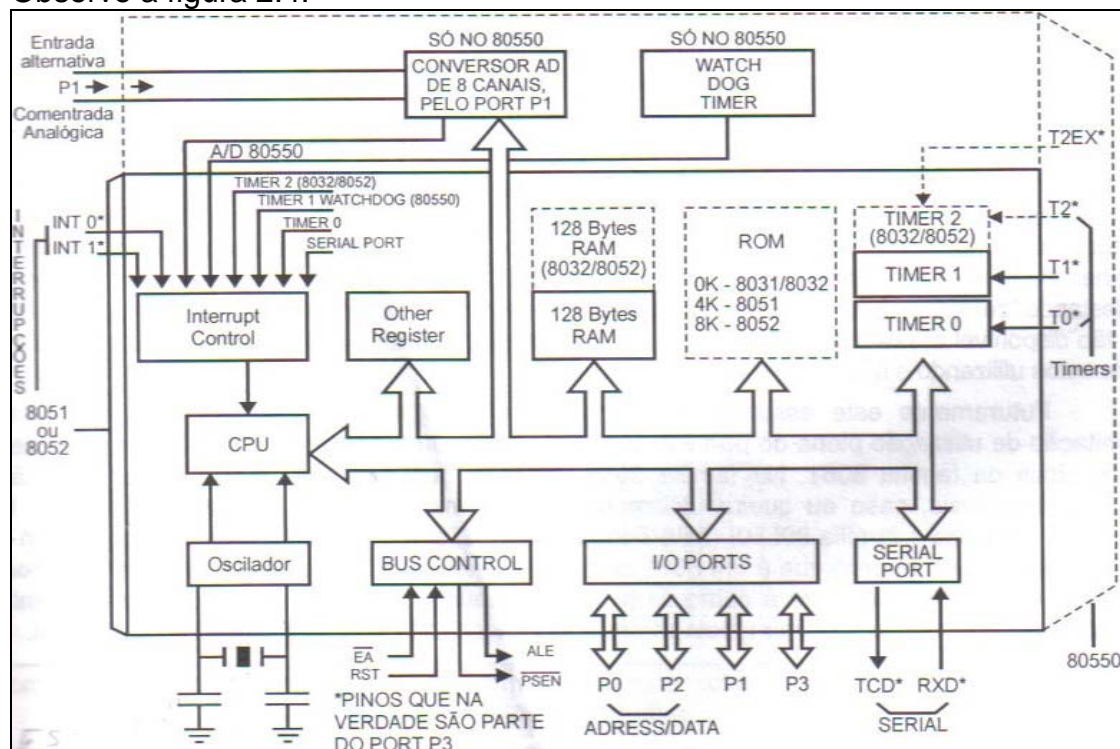


Figura 2.4 - Arquitetura interna dos microcontroladores da família 8051. [Nicolosi, 2005]

As portas PO, P1, P2 e P3, cada uma com oito linhas, são destinadas à comunicação externa. PO e P2 se destinam a gerenciar as vias de dados e endereços da comunicação do microcontrolador com a ROM, RAM ou periféricos tipo "I/O mapeado". P1 e P3 se destinam às vias de comunicação externa, sendo tipicamente usadas para interface com o mundo externo. [Nicolosi, 2005]

Além disto, a porta P3 tem funções especiais por onde se comunicam os periféricos internos, que existem nos microcontroladores: timers (2 deles no 8051, 3 deles no 8052), serial (1 unidade) e A/D (de 8 canais, somente no 80550, que neste caso usa a porta P1 como 8 entradas analógicas, além de ter normalmente os timers e serial do 8051). Os asteriscos na figura acima estão justamente representando que estes pinos não são separados dos ports P1 e P3. [Nicolosi, 2005]

Por exemplo, TXD* e RXD* são os pinos P3.0 e P3.1, isto é, os pinos "zero" e "um" da porta P3. O mesmo acontece com T2EX*, T2*, T1* e TO*. Eles fazem parte das "portas" P3 e P1. Logo, a porta P3 não é tão disponível assim como aparenta. A porta P1 é plenamente utilizável como porta de 8 vias nos chips da família 8051. Na família 8052 os pinos P1.0 e P1.1 são compromissáveis, caso seja necessário utilizar o terceiro timer que este chip tem disponível. O mesmo se aplica à família 80550, (este é igual ao 8051, só que a porta P1 é compatível com a utilização do A/D interno) que é um 8051 com um A/D de 8 canais dentro do chip de 40 pinos. Na figura 2.5 é apresentada a visão externa do chip. [Nicolosi, 2005]

Chip Externamente (Esquema elétrico)

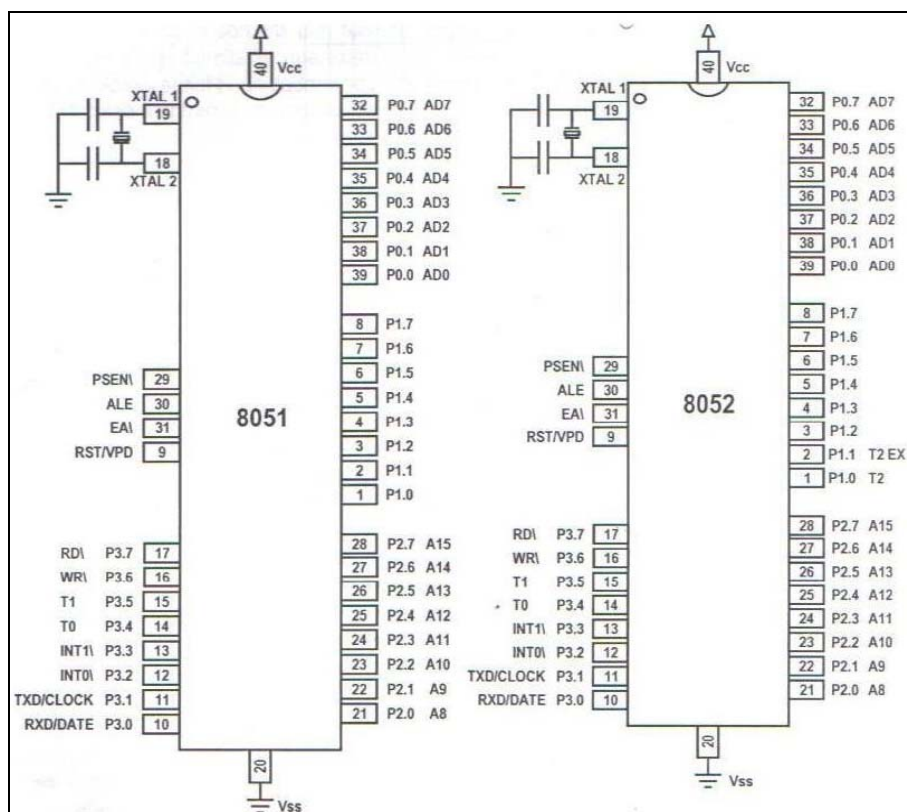


Figura 2.5 - Desenho externo do chip (família 8051 e família 8052). [Nicolosi, 2005]

2.7 ORGANIZAÇÃO DAS MEMÓRIAS

Esse processador tem um espaço separado para memória de programas e para memória de dados. Nesses microcontroladores, existe memória RAM interna, além de poder gerenciar RAM externa. O mesmo acontece com a ROM/EPROM. Na figura 2.6 [Nicolosi e Bronzeri, 2005]

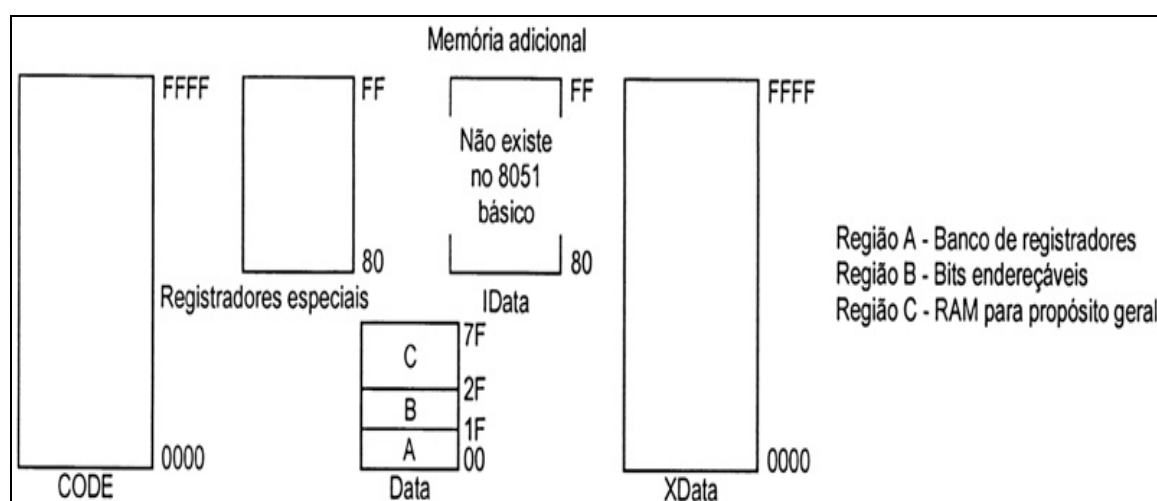


Figura 2.6 - Diagrama em bloco representando os tipos de memória. [Nicolosi e Bronzeri, 2005]

2.7.1 MEMÓRIA PARA CÓDIGO (CODE)

Esta é a região de memória em que está alojado o programa executável. Em representantes como o 80C5 I, temos 4K de memória ROM para essa finalidade. A versatilidade nos modelos está diretamente relacionada com a forma de armazenamento do programa a ser executado. Modelos mais atuais, como AT89S8252, possuem um tipo de memória capaz de ser reprogramada eletricamente. A memória destinada a essa função é denominada de Flash. Com ela é possível fazer o download do

programa gerado, de até 8 Kbytes, para esse referido "chip". [Nicolosi e Bronzeri, 2005]

Além de armazenar as instruções a serem executadas, as memórias do tipo ROM podem armazenar dados fixos cujo programa acessará. Com isso podemos armazenar tabelas de dados que serão sempre consultadas. Note que para esse tipo de memória podemos apenas fazer a leitura dos dados. Não podemos manipulá-los nem salvar dados que foram manipulados durante a execução do programa. [Nicolosi e Bronzeri, 2005]

Uma outra técnica de armazenamento do programa executável é utilizarmos uma memória SRAM, ou RAM estática em vez de utilizarmos uma EPROM. Com essa técnica, caso a alimentação seja desligada, perde-se o programa que fora armazenado, havendo a necessidade de ser gravado novamente. [Nicolosi e Bronzeri, 2005]

2.7.2 ESPAÇO PARA MANIPULAÇÃO DE DADOS (DATA)

Essa região de memória é destinada à manipulação dos dados. Para que isso ocorra, é utilizada uma memória do tipo RAM. [Nicolosi e Bronzeri, 2005]

Diferente da memória ROM, a RAM armazena temporariamente os dados, servindo, por exemplo, para salvarmos resultados de equações calculadas temporariamente. [Nicolosi e Bronzeri, 2005]

No diagrama em bloco apresentado na tabela 2.2, temos 128 bytes de memória RAM. Modelos como o AT89S8252, que futuramente será explorado, apresenta extensão de memória RAM (região IDATA), chegando a ter 256 bytes de memória RAM interna. [Nicolosi e Bronzeri, 2005]

Tabela 2.2 – Região “DATA” da memória

	Intervalo de Endereço (Hex)	Nome	Observações
Região C	7F ↑ 30	Byte endereçável	Posições de memória acessíveis apenas por byte
	2F ↑ 20	Byte endereçável	Posições de memória acessíveis por bit e por byte
Região B	1F ↑ 18	R7 BANCO3 R0	Registradores em grupo de 4 bancos
	17 ↑ 10	R7 BANCO2 R0	Endereçamento direto
	0F ↑ 08	R7 BANCO1 R0	Endereçamento indireto
	07 ↑ 00	R7 BANCO0 R0	Endereçamento pelo Stack Pointer
Região A			

Essa área de memória é dividida em três regiões. A primeira região possui quatro grupos, e cada um possui oito registradores de 8 bits que vão de R0 a R7. Já as regiões B e C são diferenciadas pelo modo de acesso aos bits dos registradores. No bloco C, apenas temos acesso a bytes, enquanto no bloco B, além de acessarmos bytes, podemos acessar bits. [Nicolosi e Bronzeri, 2005]

Os bancos de registradores são selecionados por meio de dois bits situados no registrador especial PSW (Processar Status Word). Futuramente, apresentaremos com maiores detalhes esse registrador. [Nicolosi e Bronzeri, 2005]

2.7.3 REGISTRADORES PARA FUNÇÕES ESPECIAIS (SPECIAL FUNCTION REGISTER-S.T.R.)

Os registradores especiais estão presentes em uma área da memória RAM interna que tem início no endereço 80h e término no endereço FFh. Cada registrador é composto por 8 bits, e na maioria dos casos, cada bit possui uma função especial, vinculado sempre a alguma função do microcontrolador. [Nicolosi e Bronzeri, 2005]

Com esses registradores temos condições de utilizar todos os recursos disponíveis no 8051. A tarefa de relacionar os nomes dos registradores e seus específicos endereços é feita pelo compilador C, tendo cada microcontrolador uma biblioteca de sintaxe que relaciona os registradores especiais com seu endereço na memória. [Nicolosi e Bronzeri, 2005]

2.7.4 EXTENSÃO PARA MEMÓRIA RAM (IDATA)

Alguns representantes da família 8051, como o AT89S8252, apresentam maior quantidade de memória RAM interna, proporcionada pela região IDATA. Para não gerar conflito com a área destinada aos registradores especiais, os dados só podem ser referenciados de forma indireta, uma vez que ambas as regiões de memória possuem os mesmos endereços. Por essa razão, um acesso à região IDATA é mais demorado do que à região DATA, justamente por necessitar de maior número de instruções para leitura e escrita. [Nicolosi e Bronzeri, 2005]

2.7.5 EXTENSÃO DE MEMÓRIA DE DADOS EXTERNA (XDATA)

Intitulada área para memória externa, sua implementação é equivalente à de um periférico. Normalmente, para essa região usa-se uma memória SRAM (RAM estática). Seu acesso pode ser feito pelo DPTR (Data Pointer Register) ou por registradores de forma indireta. Por necessitar de maior número de ciclos de máquina para ser acessada, temos novamente melhor eficiência com a utilização da região DATA (interna) para manipulação dos dados. [Nicolosi e Bronzeri, 2005]

2.8 O MICROCONTROLADOR AT89S8252 – ATMEL

Por ser um representante da família 8052, esse microcontrolador apresenta maior quantidade de memória RAM e mais um timer/counter de 16 bits que a família 8051 básica. Um dos pontos que o torna compatível com o 8051 é a região em que estão alojados os registradores especiais, trabalhando assim com palavras de 8 bits. Suas principais características são:

- 8 Kbytes de memória Flash - reprogramável pelo sistema serial de download de programas no formato binário (SPI - Serial Program Interface).
- 2 Kbytes EEPROM.
- Operação estática: 0Hz até 24MHz.
- 382x8-bit de Memória RAM Interna, sendo 256x8 bytes para os registradores especiais e as regiões A, B e C, e mais 127x8 bytes de extensão, acessados de forma indireta.
- 32 I/O ports programáveis.
- Três 16 bits timer/counters.
- Seis interrupções vetoradas.
- Porta serial programável - UART.
- SPI - Interface serial para reprogramação.
- Dois DPTRs - Data Pointers.

- Por meio da interface serial SPI, após compilarmos o programa, podemos transferi-lo para a região de memória CODE, que nesse microcontrolador é composta por 8 Kbytes de memória Flash. [Nicolosi e Bronzeri, 2005]

O microcontrolador AT89S8253 possui características idênticas ao microcontrolador AT89S8252 exceto a quantidade de memória flash. O AT89S8253 possui 4 KB a mais de memória flash, ou seja, 12KB.

Devido principalmente as facilidades que este tipo de memória proporciona, por apresentar as mesmas características citadas em parágrafos anteriores e possuir uma relação custo benefício melhor que o AT89S8252 este será o microcontrolador utilizado para a realização do projeto.

Utilizar o microcontrolador AT89S8253 não causa prejuízo algum para a implementação do projeto e ainda possibilita a vantagem de possuímos 4KB a mais de memória flash em caso de necessidade e prevendo uma possível expansão. Apesar de pertencer a família 8052 este chip é perfeitamente compatível com a família 8051 conforme informado acima.

CAPÍTULO 3. PROTÓTIPO

3.1 INTRODUÇÃO.

Neste capítulo são apresentados os seguintes pontos: topologia inicial, as principais especificações do Kit de gravação do microcontrolador, linguagem e ferramentas de programação utilizadas, o porquê da utilização de tais ferramentas, o funcionamento da interface receptora do sinal DTMF, que possui como principal elemento o receptor DTMF modelo MT8870. São descritas também as atribuições do microcontrolador AT89S8253 no protótipo desenvolvido. As especificações completas, os “datasheets” dos dois principais componentes do projeto, receptor DTMF e o microcontrolador 89S8253, estarão em anexo.

Na figura 3.1 é possível visualizar a topologia do projeto desenvolvido.

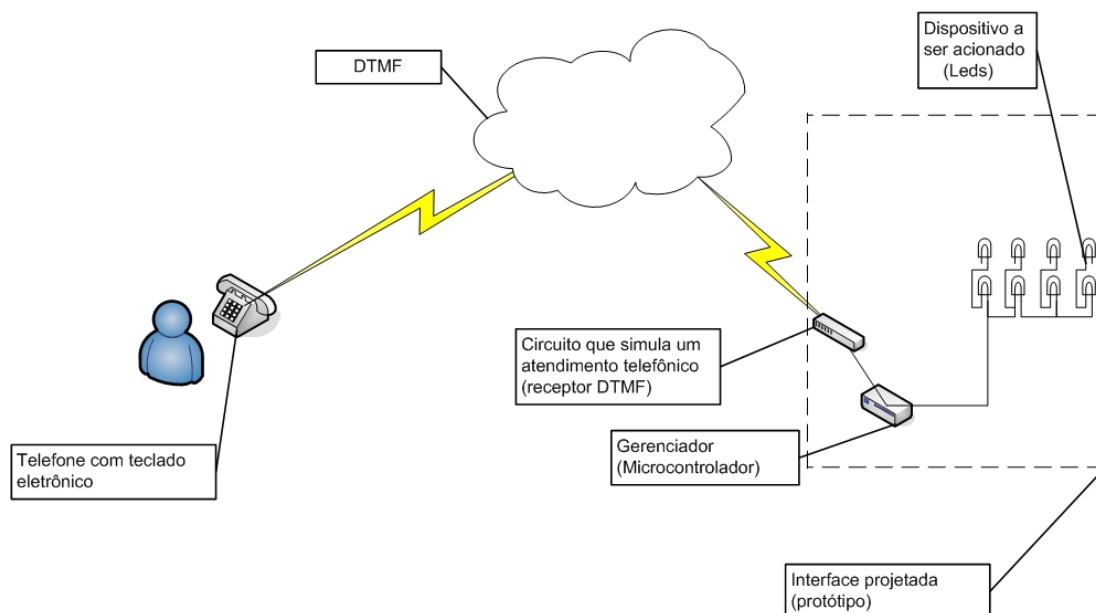


Figura 3.1 - Topologia do projeto “ Acionamento remoto de dispositivos elétricos utilizando tecnologia DTMF e Microcontroladores”.

Com esta topologia apresentada estaremos realizando todos objetivos listados na proposta inicial. Relembrando o que foi descrito no primeiro capítulo podemos resumir da seguinte forma:

- O usuário disca para o dispositivo através do teclado do telefone.
- A parte do protótipo responsável por simular um atendimento telefônico é acionada.
- O usuário recebe uma confirmação deste “atendimento” e digita no teclado a hora e a data que o dispositivo deve ser acionado.
- A parte da interface composta pelo microcontrolador recebe todas os dados enviados pelo circuito que simula um atendimento telefônico e gerencia as ações que devem ser tomadas.
- Se o dispositivo (Led) acender ou não corretamente o usuário recebe um retorno informando que o evento ocorreu ou não com sucesso.

A seguir, apresentaremos detalhadamente todos os itens envolvidos na construção da interface projetada, estabelecendo no decorrer do capítulo uma relação com a topologia apresentada anteriormente.

3.2 INTERFACE RECEPTORA DO SINAL DTMF.

Esta parte do protótipo será responsável por receber os sinais DTMF enviados pelo telefone, validar os pares de frequências recebidos e decodificá-los em binário para envio ao microcontrolador. Dividimos esquematicamente o circuito em 2 partes interligadas que serão descritas a seguir.

3.2.1 PARTE I: DETECÇÃO DE RING E COMUTAÇÃO.

3.2.1.1 DETECÇÃO DE RING.

O “ring” pode ser caracterizado por um sinal de toque (enviado pela central telefônica), na forma de uma onda AC (corrente alternada). Embora a frequência possa variar entre 15 e 68 Hz, a maior parte do mundo utiliza frequências entre 20 e 40 Hz. A tensão deste sinal está caracterizada por -48V d.c. (corrente contínua) que é gerada através de baterias localizadas nas centrais telefônicas.

[<http://www.tkk.fi/Misc/Electronics/circuits/teleinterface.html#general>, 2007]

A duração de 1 (um) ring é de cerca de 1 (um) segundo e pode variar de país para país e até mesmo de operadora para operadora. [<http://www.tkk.fi/Misc/Electronics/circuits/teleinterface.html#general>, 2007]

Esta parte do circuito é responsável por detectar o “ring”, ou seja, verificar se alguém está realizando uma chamada telefônica para este número. [<http://www.tkk.fi/Misc/Electronics/circuits/teleinterface.html#general>, 2007]

Na figura 3.2 é verificado a parte do circuito responsável pela detecção de ring e comutação.

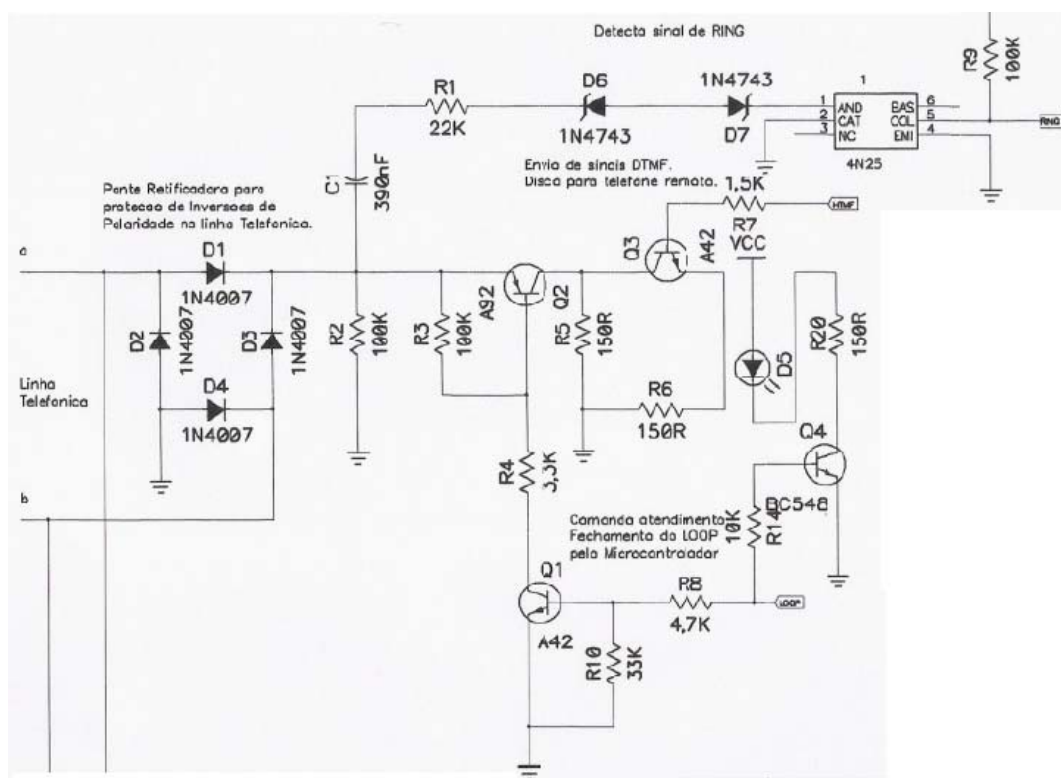


Figura 3.2 – Parte do circuito responsável pela detecção de ring e comutação.

3.2.1.2 COMUTAÇÃO.

Para permitir que uma ligação seja realizada, após a chamada telefônica (ring), o próximo passo é “atender” esta ligação. O atendimento da ligação nada mais é do que retirar o telefone do gancho.

Quando o telefone está no gancho - sobre o estado "TIP" está em cerca 0v, enquanto o "RING" é de cerca de - 48v com relação à terra. Quando retira-se o telefone do gancho, ocorrem variações nas tensões dos dois estados, “Tip” e “Ring”. Um “TIP” fora do gancho possui -20v e o “Ring” cerca de -28v. Isto significa que há cerca de 8V de tensão entre os fios passando pelo telefone em condições normais de operação. [http://www.tkk.fi/Misc/Electronics/circuits/teleinterface.html#general, 2007]

Na parte do circuito utilizada para realizar a comutação ocorre a simulação de um atendimento telefônico. Isso será feito ao fechar o circuito que encontra-se em “loop”, ou seja, retirar o gancho do telefone e conseqüentemente realizar as ações descritas acima.

3.2.2 PARTE II: RECEPTOR DTMF.

Esta é a parte principal do circuito projetado. A sua função é validar a frequência recebida através dos tons DTMF e decodificá-los em binário para que sejam levados até o microcontrolador.

A configuração necessária sugerida pelo fabricante do CM8870 para detectar frequências DTMF, é a esquematizada na figura 3.2 abaixo. Foram realizadas algumas alterações no modelo sugerido para melhor ajuste as necessidades do projeto . As alterações poderão ser verificadas na figura (x) que ilustra de forma completa o protótipo da interface desenvolvida.

Na figura abaixo a frequência entra através do capacitor cerâmico de 100nF e prossegue pelos resistores de 100k até os pinos 1 e 3 do MT8870. Quando uma frequência é reconhecida, o CM8870, converte em um valor binário e disponibiliza-o nos pinos 11,12,13 e 14. Após isso, o MT8870 ativa o pino 15, mantendo-o em nível alto por alguns milisegundos. Depois esse pino retorna ao nível baixo.

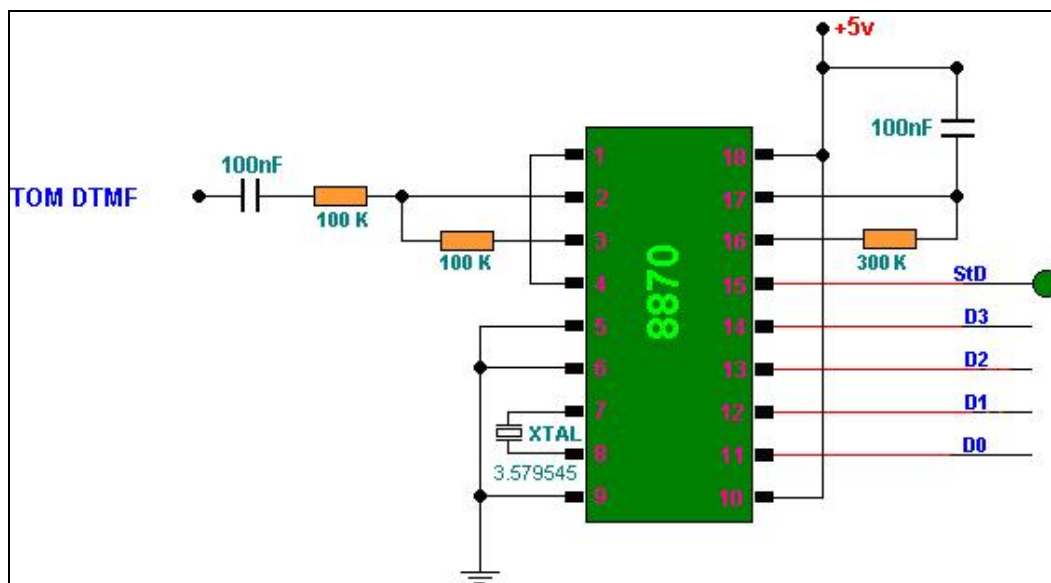


Figura 3.3 - Configuração sugerida pelo fabricante para o receptor DTMF MT8870.

A tabela 3.1 mostra o valor decimal, binário e a tecla correspondente à sobreposição das frequências DTMF (baixa e alta). [www.rogercom.com/pparalela/Lpt-ID.htm,2007]

O CI MT8870 reconhece uma frequência DTMF e converte-a em um número binário de 4 bits (1 nibble). O software aplicativo é o responsável em converter esse número binário em uma tecla equivalente ao sistema DTMF, conforme mostra a primeira coluna da tabela acima. Observe que o valor decimal **10** representa a tecla "0". [www.rogercom.com/pparalela/Lpt-ID.htm, 2007]

Tabela 3.1 - Frequências DTMF

Tecla	Saída digital				Valor Decimal	Baixa Frequência(Hz)	Alta Frequência(Hz)
	D3	D2	D1	D0			
1	0	0	0	1	1	697	1209
2	0	0	1	0	2	697	1336
3	0	0	1	1	3	697	1477
4	0	1	0	0	4	770	1209
5	0	1	0	1	5	770	1336
6	0	1	1	0	6	770	1477
7	0	1	1	1	7	852	1209
8	1	0	0	0	8	852	1336
9	1	0	0	1	9	852	1477
0	1	0	1	0	10	941	1336
*	1	0	1	1	11	941	1209
#	1	1	0	0	12	941	1477
A	1	1	0	1	13	697	1633
B	1	1	1	0	14	770	1633
C	1	1	1	1	15	852	1633
D	0	0	0	0	0	941	1633

As saídas D0,D1,D2,D3 serão conectadas ao microcontrolador AT89S8252. Na figura 3.4 é apresentada a parte do circuito responsável pela recepção dos tons DTMF.

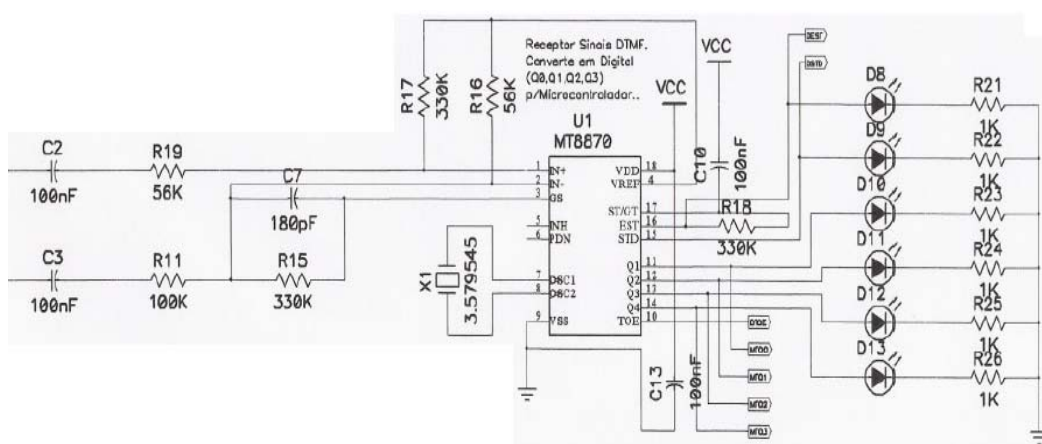


Figura 3.4 - Circuito detalhado do receptor DTMF MT8870.

3.3 MICROCONTROLADOR .

Nesta parte do projeto ocorre o gerenciamento do hardware projetado. Os sinais decodificados através do receptor DTMF são recebidos pelo Microcontrolador que de acordo com a programação inserida através do teclado eletrônico executa resumidamente as seguintes ações:

- Recebe a programação de Data e hora que o dispositivo deve ser acionado;
- Valida se a programação foi inserida com sucesso;
- Executa o acionamento do dispositivo elétrico no horário programado;
- Envia um retorno, através de uma chamada telefônica com 3 (três) toques, para o usuário indicando assim que o evento programado ocorreu com sucesso.

Na figura 3.5 é apresentada com detalhes como o microcontrolador gerencia todo o circuito.

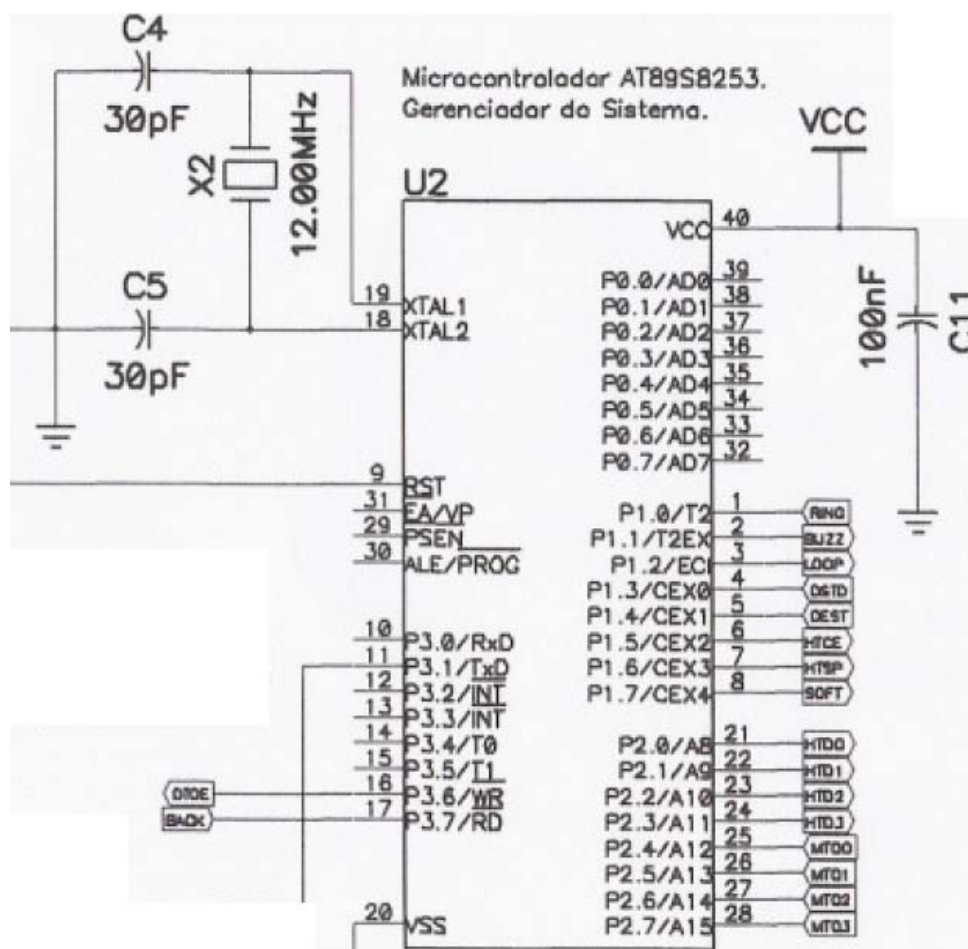


Figura 3.5 - Circuito detalhado do microcontrolador AT89S8253.

3.4 KIT AT89S8252-ATMEL (FAMÍLIA 8051).

Este hardware foi utilizado para gravar o microcontrolador com as rotinas necessárias para comandar todas as ações realizadas pela interface desenvolvida. O Kit é desenvolvido e comercializado pela empresa NEUROTREND - Technology Solutions Ltda. Este equipamento possui as seguintes características:

- Permite comunicação com o PC via serial e também via Paralela do PC, neste caso para gravação do Microcontrolador AT89S8252;

-Gravação ISP – In System Programming (programação em sistema) permite programar o microcontrolador AT89S8252 diretamente. Isto é usado para programar o “bootloader” no processador antes que a placa possa ser utilizada;

-Chave de gravação (Load/Run) com “leds” para facilitar os processos de gravação e execução do programa desenvolvido, sem precisar retirar cabos de comunicação “ISP” ou o serial;

-8 portas de entrada e saída (I/O) mapeadas para utilizar e gravar linhas de programação conforme descrito em “mapas de endereços” no manual Kit de Microcontrolador AT89S8252 – Atmel (família 8051), em anexo.

-8 KB de Flash;

-2 KB EEPROM;

-4 portas disponíveis com Conector;

- Inclusa comunicação serial RS232 que é um padrão para troca em série de dados binários entre um terminal de dados (DTE - Data Terminal equipment) e um comunicador de dados (DCE - *Data Communication equipment*). Este padrão é comumente utilizado nas portas seriais dos PCs;

O sistema apresentado é extremamente flexível, pois todas as portas do microcontrolador estão disponíveis ao usuário o que facilita estender as características da placa de acordo com a necessidade.

O sistema apresentado permite comunicação por técnica conhecida como entrada e saída (I/O) mapeada, podendo estender as portas deste micro até 8 saídas endereçadas com endereço fixo (8000h até 8007h), usando para isso as portas P0 e P2 parcialmente, e os pinos Rd e **WR** da porta P3.

Permite também executar o programa desenvolvido em uma memória Flash do AT89s8252.

Esta placa contém um microcontrolador de 8 bits (AT89S8252) que possui internamente 5 interrupções mascaráveis, 2 tempos (timers) de 16 bits programáveis e um canal serial, que já tem dentro dele uma Flash de 8KB e EEPROM de 2KB. O diagrama apresentado na figura 3.6 demonstra os principais componentes e fluxo de dados:

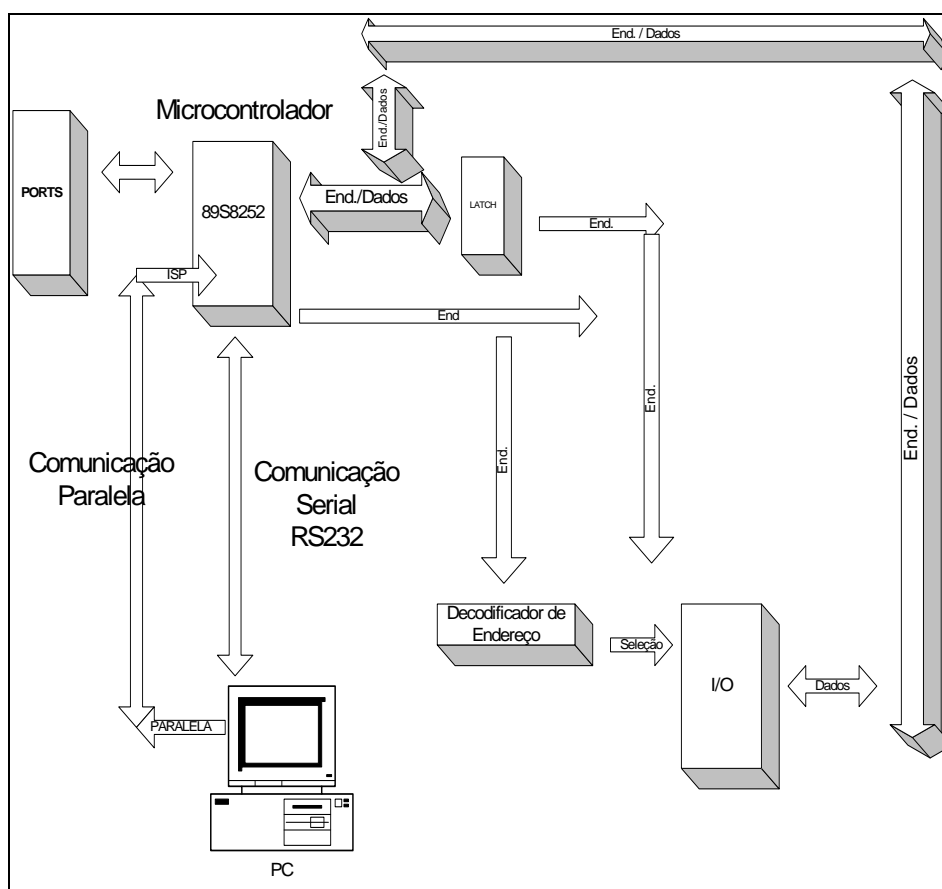


Figura 3.6 - Fluxo de dados do Kit de gravação de microcontrolador AT89S8252 – Atmel

Na figura 3.7 temos uma foto real do Kit de gravação de microcontrolador AT89S8252 –Atmel desenvolvida pela empresa NEUROTREND - Technology Solutions Ltda.



Figura 3.7 - Kit de gravação de microcontrolador AT89S8252 – Atmel

3.5 SOFTWARES DE COMPILAÇÃO E DE GRAVAÇÃO DO MICROCONTROLADOR.

3.5.1 O COMPILADOR SDCC.

O SDCC (Small Device C Compiler) é um compilador de domínio público, voltado à programação de microcontroladores que trabalhem com palavras de 8 bits. Sua função é traduzir o código criado em C para uma linguagem de montagem suportada pelo processador e fornecida pelo fabricante. Dentre as famílias suportadas por ele, temos: a família 8051, Zilog Z80, variações do Dális DS80C390 e PIC. [Nicolosi e Bronzeri, 2005]

O processo de compilação é complexo. Basicamente, o código gerado em C é compilado para um código relativo em "assembly", com extensão ".asm". A partir do código gerado em linguagem de máquina (assembly), o montador (assembler) fica responsável por criar arquivos que podem ser transferidos ao microcontrolador, e assim executados. Observe o programa seguinte:

```
#include<at89s8252.
```

```
main (void) { int a;
```

```
/*biblioteca de sintaxe relacionando os registradores especiais*/
```

Um programa em C possui alguns elementos indispensáveis como os colchetes "{" e o ponto-e-vírgula ";". Além disso, todo programa deve conter um bloco principal, denominado "main", cujo processamento do programa terá início. Esse bloco principal é envolvido por colchetes. Essa característica não se restringe apenas ao bloco principal. Caso existam outros blocos de código no programa, eles devem vir sempre limitados por colchetes. Observe a forma como é feito um comentário no programa. [Nicolosi e Bronzeri, 2005]

Esse trecho compreendido por "/* */" é ignorado no momento da compilação. Não necessariamente deve aparecer em apenas uma linha. Um comentário pode ser escrito em quantas linhas forem necessárias, havendo apenas a necessidade de estar envolvido por "/* */". [Nicolosi e Bronzeri, 2005]

Outro elemento importante é a variável, representada no "int a;". A parcela "int" que antecede o nome da variável (no caso deste exemplo nomeada "a") indica a quantidades de bits a serem utilizados pela variável. Portanto, temos uma variável do tipo inteiro (int) intitulada "a". O "ponto-e-vírgula" representa fim da sentença e é usado constantemente. Não possuem função adicional que não seja esta. Seu uso é obrigatório e caso ignorado, será indicado um erro na compilação. Estas explicações básicas são necessárias para melhor entendimento do código que foi compilado. [Nicolosi e Bronzeri, 2005]

A biblioteca de sintaxe é responsável por ter, no caso do compilador SDCC os endereços dos registradores especiais e alguns bits especiais que podem ser diretamente acessados. A ligação entre o arquivo do programa gerado que contém esses endereços é feita antes da compilação do

programa. Esses arquivos possuem extensão ".h" e são incluídos no programa pela diretiva de pré-processamento "#include". [Nicolosi e Bronzeri, 2005]

3.5.2 GRAVA ATMEL

Software gravador ISP – In System Programming (programação em sistema) para microcontroladores Atmel 89s8252, 89S8253, 89s53, 89s52 e 89s51. É compatível com o “software” Blast8252 e funciona praticamente em todas as versões do Windows. [<http://www.telegraf.cz/jak/prog.html>, 2007]

Este “software” possui teste de compatibilidade aprovado com as seguintes versões do sistema operacional Windows:

- 9x, ME, 2000, XP, 2003. [<http://www.telegraf.cz/jak/prog.html>, 2007]

Também pode rodar no sistema operacional Linux através do emulador Wine e apresenta as seguintes características:

- Suporte para arquivos .HEX (Intel-hex).
- Ferramentas de conversão entre os formatos BIN e HEX.
- Editor hexadecimal de arquivos.
- Memorização do último modelo utilizado.
- Suporte para o modelo de microcontrolador AT89S8253 (12K).

Este software será utilizado para gravação do código compilado através SDCC no microcontrolador.

3.6 INTERFACE PROJETADA

Estas são as partes principais da interface projetada, detecção de ring e comutação, receptor DTMF e Microcontrolador 89S8252. Além das partes citadas existem algumas outras que são descritas como circuitos auxiliares. Estes circuitos são utilizados para proteger a linha telefônica contra sobrecargas (ponte retificadora), permitir a comunicação serial com o microcomputador (interface RS232), enviar tons DTMF para o telefone desejado (transmissor DTMF).

Na figura 3.8 é apresentada uma visão completa da interface projetada e dos principais componentes descritos acima.

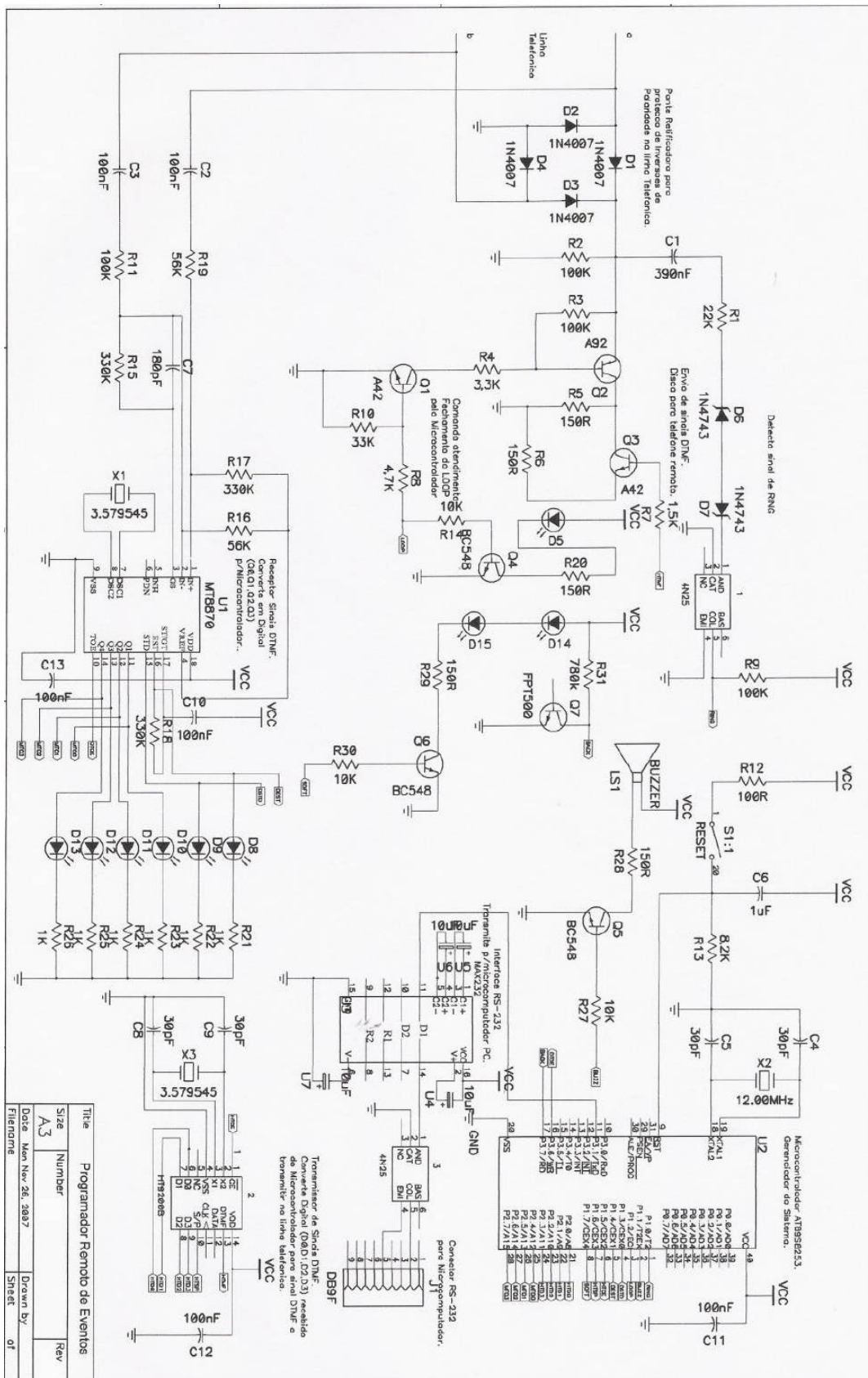


Figura 3.8 - Desenho do circuito da interface (placa) projetada.

CAPÍTULO 4. RESULTADOS E A INTERFACE PROJETADA PARA O SISTEMA

4.1 INTRODUÇÃO

O capítulo a seguir descreve todo o funcionamento do protótipo, o código fonte (comentado) da programação do microcontrolador, a descrição dos passos que o usuário deve seguir para inserir a programação e como esta ocorre. Serão apresentadas também algumas imagens que permitem a visualização de como a interface projetada realiza os eventos gerenciados pelo microcontrolador. Também serão apresentados os resultados obtidos com o projeto do sistema.

4.2 FUNCIONAMENTO DO PROTÓTIPO

Para explicar como se dá o funcionamento do projeto construído são descritos os seguintes passos:

- Inicializar o ambiente de execução, inclusive com a criação de um relógio de tempo real;
- Passar a aguardar o recebimento do sinal de RING (chamada pelo telefone remoto);
- Receber o RING e temporizar 1 segundo;
- Fechar o LOOP (simula a retirada do telefone do gancho);
- Aguardar o recebimento dos comandos, na sequência:

1DDMMAAHMMSS - Digitar (*1), data atual (DiaMêsAno-DDMMAA) e hora atual (HoraMinutoSegundo-HHMMSS). Fechar o

comando com (*).

2DDMMAAHMMSS - Digitar (*2), data de realização do evento (DiaMêsAno-DDMMAA) e hora que o evento deve ocorrer (HoraMinutoSegundo-HHMMSS). Fechar o comando com (*).

*3PREFMCDU# - Número telefônico a ser chamado pelo Programador Remoto de Eventos na data/hora programada, onde: PREF = Prefixo do número do telefone programado a chamar;

MCDU (Milhar, Centena, Dezena e Unidade do número do telefone programado a chamar).

- Ex: 3368-8821 (PREF = 3368 e MCDU = 8821);
- Após a confirmação do sucesso do recebimento de todos os comandos do item 5, abre novamente o LOOP da linha telefônica;
- O tempo começa a ser monitorado;
- Uma vez decorrido o tempo programado:

Sinalizar (LED D14 no circuito) o disparo do evento;

Discar para o número programado;

- Finalizar a operação.

Estes são os passos que precisam ser seguidos para que o protótipo funcione corretamente.

4.3 PROGRAMAÇÃO DOS EVENTOS PELO USUÁRIO

Nesta fase do trabalho estaremos descrevendo como o usuário deve

proceder para que o protótipo funcione.

O usuário deve proceder da seguinte forma:

- Informar *1DDMMAAHMMSS* - Data Atual (DiaMêsAno) e Hora atual (HoraMinutoSegundo);
- Informar *2DDMMAAHMMSS* - Data para o Evento (DiaMêsAno) e Hora para o Evento (HoraMinutoSegundo);
- Informar *3PREFMCDU# - Número telefônico a ser chamado pelo Programador Remoto de Eventos na data/hora programada, onde:

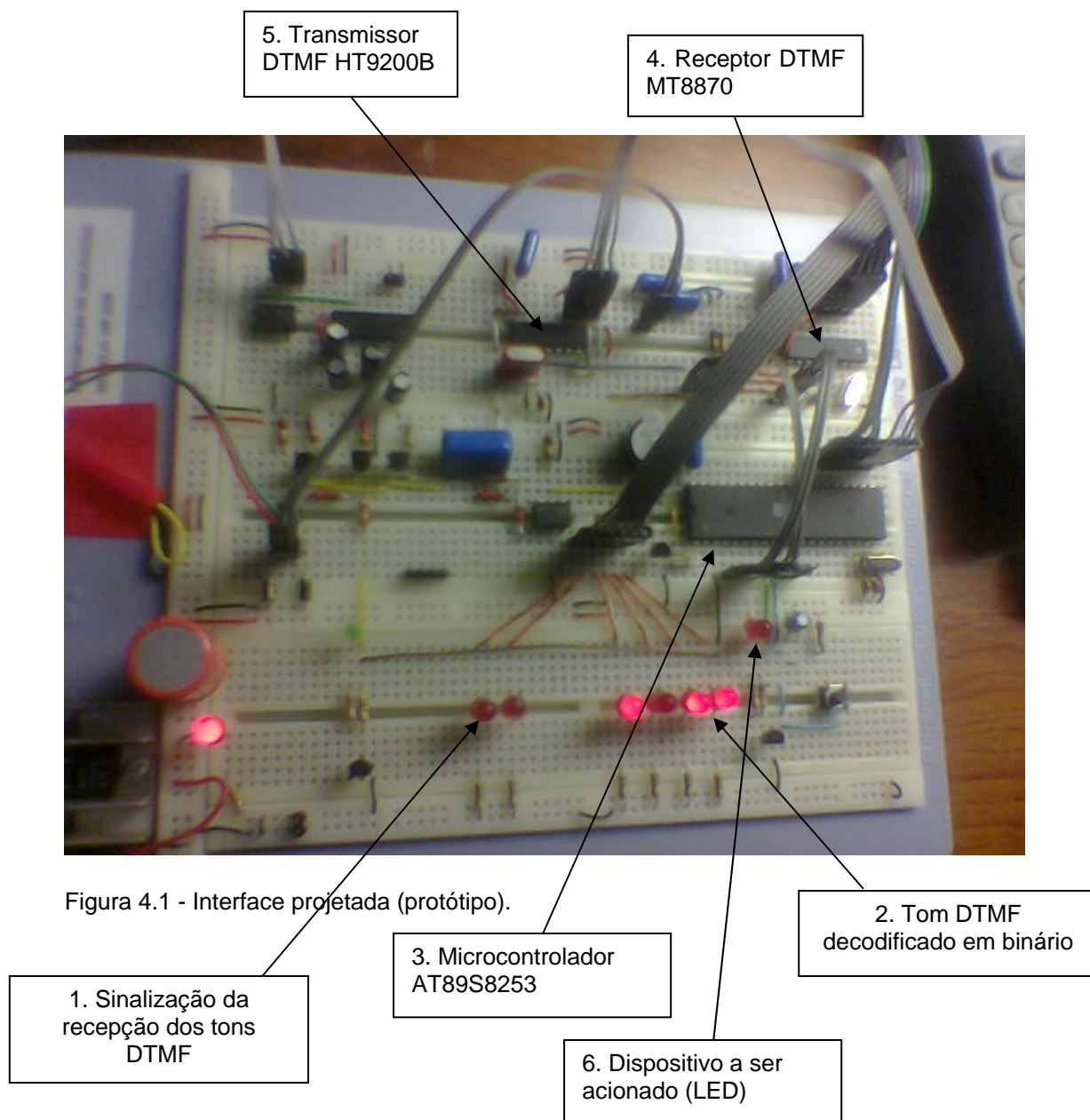
PREF = Prefixo do número do telefone programado a chamar;

MCDU (Milhar, Centena, Dezena e Unidade do número do telefone programado a chamar);

Ex: 3368-8821 (PREF = 3368 e MCDU = 8821);

Uma vez decorrido o tempo programado, o LED utilizado como o dispositivo elétrico a ser acionado acende se o evento ocorrer com sucesso. Caso não acenda, o evento não ocorreu da forma desejada. O próximo evento é realizar a discagem para o número programado.

4.4 IMAGENS DA INTERFACE PROJETADA



Observando a figura 4.1 é possível identificar as seguintes situações:

1. Neste ponto é possível visualizar através dos “LEDs” a sinalização da recepção dos tons DTMF que estão sendo enviados pelo usuário através

da linha telefônica.

2. Saída que indica os binários decodificados que são oriundos dos tons DTMF (também pode ser visualizada através dos “LEDs”).

3. O microcontrolador fica responsável por realizar o gerenciamento de todas as rotinas de controle do sistema.

4. Recebe os tons DTMF (analógico), decodifica e converte em sinal digital.

5. Faz a geração dos tons DTMF para realizar a ligação para o telefone desejado.

6. Dispositivo que deve ser acionado (LED).

Realizada a sequência de programação definida no item 4.4 é possível observar o acionamento do dispositivo remoto, neste caso o LED (item 6) indicado pela figura 4.1. Após o acionamento do LED o próximo passo é a realização da chamada para o telefone informado. Se o evento ocorrer com sucesso o telefone tocará três vezes. Se por algum motivo o dispositivo não for acionado de forma correta, o telefone tocará uma vez. Com todas essas etapas realizadas pode desta forma concluir que o protótipo está funcionando de forma correta conforme os objetivos descritos no capítulo 1 (um).

CAPÍTULO 5. CONCLUSÕES E PROJETOS FUTUROS

5.1 CONCLUSÕES

A tecnologia DTMF está bastante difundida e aprimorada em todo o mundo. É um tipo de sinalização bastante confiável e utilizada amplamente utilizada nas linhas telefônicas. Após a implementação deste projeto, percebemos que esta tecnologia poderia ser melhor aproveitada devido ao grande número de aplicações que podem utilizar este "canal" para a transmissão de dados. Em nosso país este tipo de tecnologia está destinada praticamente ao uso da telefonia. Utilizá-la para outros fins proporcionaria um melhor aproveitamento deste recurso.

Os microncontroladores da família 8051 ou compatíveis com a mesma são bastante conhecidos e utilizados em milhares de equipamentos eletrônicos. O baixo custo, pequeno consumo de energia e a confiabilidade de uma tecnologia que está consolidada foram fundamentais para a utilização deste componente na construção do protótipo apresentado. O microcontrolador AT89S8253 atendeu perfeitamente as necessidades de implementação do projeto.

Realizar um estudo detalhado e verificar através de testes o comportamento físico do hardware projetado, antes mesmo de utilizar o microcontrolador para fazer o gerenciamento do circuito, foram ações fundamentais para facilitar a programação das rotinas e garantir o funcionamento correto do protótipo. Gastar um tempo maior nesta fase do projeto permitiu uma segurança bastante satisfatória para realizarmos as outras etapas do mesmo.

Unir a sinalização DTMF e microcontroladores é com certeza um ótimo caminho para o acionamento remoto de dispositivos. Controlar remotamente um equipamento possibilita economia de tempo, menor custo operacional, agilidade e requer confiabilidade. A utilização das tecnologias

citadas apresenta a confiabilidade necessária para o controle remoto de dispositivos e por isso foram escolhidas para a realização do projeto realizado.

5.2 PROJETOS FUTUROS

Como sugestão para projetos futuros destacam-se duas propostas:

Realizar a implementação deste mesmo protótipo, mas, com o seguinte diferencial, utilizar SMS ao invés da DTMF e integrar mais esta aplicação aos telefones móveis (celulares). Isso possibilitará mais mobilidade e agilidade ao usuário e também uma interface mais amigável.

A outra sugestão seria construir uma outra interface para comunicação com o protótipo desenvolvido e assim permitir o acionamento de equipamentos de grande porte como motores, geradores, bombas d'água e outros. Com a construção desta interface, a intenção é atingir o mercado industrial e a produção em larga escala de toda a solução, ou seja, uma placa que possa controlar remotamente equipamentos elétricos de qualquer porte e que utilize como canal de comunicação a tecnologia DTMF e SMS.

5.3 DIFICULDADES ENCONTRADAS

Impossibilidade de realizar a demonstração no Centro Universitário de Brasília-UNICEUB devido a interferência causada pela configuração do PABX. Sensibilidade excessiva do receptor DTMF o que ocasionou a dificuldade de receber a programação através dos tons DTMF em alguns testes.

REFERÊNCIAS BIBLIOGRÁFICAS

DATASHEET CATALOG [Home Page]. Disponível em <<http://www.datasheetcatalog.net>; Acesso em: 20 de março de 2007

ENGDAHL, Tomi. **Telephone line audio interface circuits** [2001]. Artigo sobre detecção de ring. Acesso em: 01 jan. 2007. Disponível em: <http://www.tkk.fi/Misc/Electronics/circuits/teleinterface.html#general>

FERRARI, Antonio Martins. **Telecomunicações: Evolução & Revolução**. 9. ed. São Paulo: Erica, 2005

FERRARI, Rafael Balthazar. **Kit Didático para Microcontrolador Atmel**. [2002]. Manual do Kit de gravação AT89S8252. Acesso em: 10 jan. 2007. Disponível em: <http://www.microcontrolador.com.br/documentacao.php?&id=4>

JIRICEK, Jakub. **Atmel AT89S8252 in-system programming via PC printer port** [2001]. Software de gravação do microcontrolador AT89S8252. Acesso em: 17 mar. 2007. Disponível em: <http://www.telegraf.cz/jak/prog.html#new>

MARTINS, Roberto. **A Fundamentação da Telefonia através da História - Parte 1: Da Invenção ao Início do Século XX** [2003] pesquisa realizada para a Fundação Telefônica, em 2002). Acesso em: 01 jun. 2006. Disponível em: <http://www.museudotelefone.org.br/descoberta.htm>.

MESSIAS, Antônio Rogério. **Identificador de chamadas por sinalização DTMF** [2006]. Artigo sobre identificação de chamadas. Acesso em: 15 fev. 2007. Disponível em: <http://www.rogercom.com/>

MITEL SEMICONDUCTOR. **Applications Of The MT8870 Integrated DTMF Receiver**. Application Note MSAN-108 MITEL, 2004

NICOLOSI, Denys Emílio Campion; BRONZERI, Rodrigo Barbosa. **Microcontrolador 8051 com linguagem C: prático e didático – família AT89S8252 Atmel**. 1. ed. São Paulo: Erica, 2005

GLOSSÁRIO DE TERMOS

DTMF - Dual-tone multi-frequency

RF – Radio Frequency

EPROM – Erasable Programmable Read-Only Memory

SPI – Serial Program Interface

PSW – Processar Status Word

DTE – Data Terminal Equipment

DPTR – Data Pointer Register

DCE – Data Communication Equipment

RAM – Random Access Memory

CAN – Control Area Network

Anexo 1 – Código fonte comentado.

```
/******
```

Projeto: Sistema PROGRAMADOR REMOTO DE EVENTOS

Autor: Luiz Carlos Borges Filho
Em: nov/2007

Descrição: Módulo controlador que, a partir de dados entrados remotamente, por linha telefônica, fica programado para ativar algum evento, em data (DDMMAA) e hora (HHMMSS). O sistema, a partir da data atual fornecida, passa a incrementar o tempo, a cada segundo (através da interrupção 5 - Timer2), e por comparação da data e hora atuais incrementadas, com a data e hora futura, assegura a ocorrência do evento programado. O evento é então ativado pelo acendimento do LED D5 e pela discagem para o número telefônico a ser chamado.

Microcontrolador Usado: AT89S8253

```
*****/
```

```
/* Arquivos Header */
```

```
#include <at89s8253.h>
#include <string.h>
```

```
/* Definições das Constantes de Programação */
```

```
#ifdef (MT00)
```

```
#define MT00 P2_0; //Bit 0 do DTMF codificado pelo AT89S8253, a enviar  
pela linha telefônica
```

```
#define MT01 P2_1; //Bit 1 do DTMF codificado pelo AT89S8253, a enviar  
pela linha telefônica
```

```
#define MT02 P2_2; //Bit 2 do DTMF codificado pelo AT89S8253, a enviar  
pela linha telefônica
```

```
#define MT03 P2_3; //Bit 3 do DTMF codificado pelo AT89S8253, a enviar  
pela linha telefônica
```

```
#define HT00 P2_4; //Bit 0 do DTMF decodificado
```

```
#define HT01 P2_5; //Bit 1 do DTMF decodificado
```

```
#define HT02 P2_6; //Bit 2 do DTMF decodificado
```

```
#define HT03 P2_7; //Bit 3 do DTMF decodificado
```

```
#define DTOE P1_0; //Bit para ler sinal de RING recebido da linha telefônica
```

```

#define BUZZ P1_1; //Bit para sinalizar erros de digitação dos comandos
(BUZZ = buzzer)
#define EVT_ P1_2; //Bit p/comandar fechamento/abertura de LOOP da
linha telefônica
#define DSTD P1_3; //Bit indicador do sinal STD do MT8870
#define DEST P1_4; //Bit indicador do sinal EST do MT8870
#define HTCE P1_5; //Bit para habilitar/desabilitar o envio de sinais DTMF
pelo HT9200
#define HTSP P1_6; //Bit para configurar o chip HT9200 como paralelo
#define SOFT P1_7; //Bit para uso pelo software para diversas sinalizações

#endif

```

```

/* Definições das Variáveis Globais */

```

```

char DATUAL,DFUTUR;
char HATUAL,HFUTUR;
char NUMDISCAR[9];

```

```

char dia_atu,mes_atu,ano_atu;
char hor_atu,min_atu,seg_atu;

```

```

char dia_fut,mes_fut,ano_fut;
char hor_fut,min_fut,seg_fut;

```

```

char hor_mais,min_mais,seg_mais;
char ano_mais,mes_mais,dia_mais;

```

```

char cmd_rec[13];
char ch_rec;
int num_interv_50ms;

```

```

/* Funções do Sistema */

```

```

void inic_t2_50ms() {
    /******
    Gera um intervalo de 50 milisegundos usando interrupção do Timer2
    Cinquenta milisegundos é 0xC350 (Hexadecimal). O seu complemento de 2
    é 0x3CAF. A cada interrupção do timer 2, esta função é chamada.
    *****/
    TH2=0x3C;
    TL2=0XAF;
    T2CON=0X04; //Habilita o Timer2
}

```

```

void timer2() interrupt 5 {
    TF2=0; //Reseta o flag de overflow do timer2
}

```

```

    num_interv_50ms--;
}

/*****
    Função que executa a quantidade de 10 milisegundos, passada pela
    variável nxms
    *****/
void espera_nx10ms(char nx) {
    while (nx) {
        TMOD=0x01; //timer0 no modo 1 (16 bits)
        TH0=0xD8;
        TL0=0xEF;
        TR0=1; //Inicia a contagem
        TF0=0;
        while(!TF0); //Aguarda fim da contagem de tempo (50 milisegundos)
        nx--;
    }
}

/*****
    Função que executa a quantidade de 50 milisegundos, passada pela
    variável nxms
    *****/
void espera_nx50ms(char nx) {
    while (nx) {
        TMOD=0x01; //timer0 no modo 1 (16 bits)
        TH0=0x3C;
        TL0=0xAF;
        TR0=1; //Inicia a contagem
        TF0=0;
        while(!TF0); //Aguarda fim da contagem de tempo (50 milisegundos)
        nx--;
    }
}

/*****
    Função que configura a porta serial para transmissão
    *****/
void cfg_serial() {
    TMOD=0x20; //Modo de operação com timer1
    SCON=0x40; //Modo de Operação sem recepção
    TCON=0x40; //Habilita contagem pelo timer1
    TH1=0xFD; //Baud Rate de 9600
    TL1=1;
}

/*****
    Função que Transmite um caracter

```

```

*****/
void tx(char *btx)  {
    cfg_serial();
    SBUF=*btx;
    while(!TI);
}

/*****
    Função que monta os 4 bits do dtmf recebido
*****/
char dtmf_rec() {
    char bit0=P2_4;
    char bit1=P2_5;
    char bit2=P2_6;
    char bit3=P2_7;
    char drec=0x00;
    drec=drec|(bit3<<3);
    drec=drec|(bit2<<2);
    drec=drec|(bit1<<1);
    drec=drec|(bit0);
    return drec;
}

int valida_decimal(char ch) {
    if ((ch>=0x0)&&(ch<=0x9)) {
        return 1;
    }
    else {
        return 0;
    }
}

void buzz(int x)    {
    while(--x>=0){
        P1_1=1; //Ativa buzzer (sonoriza falha)
        P1_7=1;  //Ativa sinalização visual (LED)
        espera_nx50ms(2);
        P1_1=0; //Desliga buzzer
        P1_7=0;  //Desativa sinalização visual (LED)
        espera_nx50ms(2);
    }
}

void beep_tx(int nb) {
    P3_6=0;      //Desabilita o receptor de tons DTMF (MT8870)
    while(--nb>=0)    {
        P2=0x07;
        P1_5=0;      //Habilita o gerador de tons DTMF
    }
}

```



```

        espera_nx50ms(6);
        P1_5=1;      //Desabilita o gerador de tons DTMF
        espera_nx50ms(6);
    }
    P3_6=1;      //Re-Habilita o receptor de tons DTMF (MT8870)
}

void recebe_programacao() {
    char *p;
    char cmd_atu_rec;
    int k;
    int num_cmd_rec=0;
    int pos_car_rec=1;
    int prog_completa=0;

    while(prog_completa==0) {

        for (k=0; k<16; k=k+1) { //Zera (limpa) o array de recebimento de
comandos
            cmd_rec[k]=0x00;
        }

        while(num_cmd_rec<3) {
            if ((P1_3==1)&&(P1_4==1))      //Se sinais DSTD e DEST
foram detectados (tom DTMF presente)
                ch_rec=dtmf_rec();
                if (ch_rec==0x0A) ch_rec=0x00;
                if (ch_rec==0x0B)  { //Recebeu caracter "*"
                    if (pos_car_rec==1) { //Abertura de comando
                        pos_car_rec++;
                        buzz(1);
                        //beep_tx(1);
                    }
                    else if (pos_car_rec==15) { //Fechamento de
comandos 1 ou 2
                        if (cmd_atu_rec==1){ //Monta data e hora
atuais
                            dia_atu=cmd_rec[0]*10+cmd_rec[1];

                            mes_atu=cmd_rec[2]*10+cmd_rec[3];

                            ano_atu=cmd_rec[4]*10+cmd_rec[5];
                                hor_atu=cmd_rec[6]*10+cmd_rec[7];

                            min_atu=cmd_rec[8]*10+cmd_rec[9];

                            seg_atu=cmd_rec[10]*10+cmd_rec[11];
                        }
                    }
                }
            }
        }
    }
}

```

```

else    //{Monta data e hora futuras
        dia_fut=cmd_rec[0]*10+cmd_rec[1];

mes_fut=cmd_rec[2]*10+cmd_rec[3];
        ano_fut=cmd_rec[4]*10+cmd_rec[5];
        hor_fut=cmd_rec[6]*10+cmd_rec[7];
        min_fut=cmd_rec[8]*10+cmd_rec[9];

seg_fut=cmd_rec[10]*10+cmd_rec[11];
    }
    for (k=0; k<16; k=k+1) //{Zera (limpa) o
array de recebimento de comandos
        cmd_rec[k]=0x00;
    }
    pos_car_rec=1;    //{Renicializa o ponteiro
da posição do caracter
        num_cmd_rec++;
        buzz(3);
        //beep_tx(3);
    }
    else    //{Caraceter inválido para a posição
        buzz(2);
        //beep_tx(2);
    }
}
else if (ch_rec==0x0C)    //{Recebeu caracter "#"
    if (pos_car_rec==11)    {
        p=&NUMDISCAR;
        for (k=0; k<8; k++) {
discagem DTMF
            if (cmd_rec[k]==0x00)    //{Recupera o zero para
                cmd_rec[k]=0x0A;
            }
            *p++=cmd_rec[k];
        }
        buzz(4);
        //beep_tx(4);
        prog_completa=1;
        num_cmd_rec++;
    }
    else    //{Caracter inválido para a posição
        buzz(3);
        //beep_tx(3);
    }
}
else    //{Recebeu caracter decimal
    if (pos_car_rec==2) //{É identificador do comando
        cmd_atu_rec=ch_rec;

```

```

        buzz(1);
        //beep_tx(1);
    }
    else { //Copia caracter recebido, na posição,
para Array de comandos recebidos (cmd_rec)
        cmd_rec[pos_car_rec-3]=ch_rec;
        buzz(1);
        //beep_tx(1);
    }
    pos_car_rec++;
}
}
} //while(num_cmd_rec<3)
} //while(prog_completa==0);
}

void mostra_segundos() {
    P0_0=seg_atu&0x01;
    P0_1=(seg_atu>>1)&0x01;
    P0_2=(seg_atu>>2)&0x01;
    P0_3=(seg_atu>>3)&0x01;
    P0_4=(seg_atu>>4)&0x01;
    P0_5=(seg_atu>>5)*0x01;
    P0_6=(seg_atu>>6)&0x01;
}

void monitora_evento() {
    while(1) {
        if (num_interv_50ms==0) { //Completo intervalo de 1 segundo
            P0=seg_atu+1;
            num_interv_50ms=20;
            if (++seg_atu>59) { //Incrementa o segundo atual
                seg_atu=0;
                if (++min_atu>59) {
                    min_atu=0;
                    if (++hor_atu>23) { //Passou para a Meia-
noite
                        hor_atu=0;
                        dia_atu++; //Incrementa o dia
                        switch (mes_atu) {
                            case 1:
                                if (dia_atu>31) {
                                    dia_atu=1;
                                    mes_atu=2;
                                }
                                break;
                            case 2:
                                if (ano_atu%4==0)

```

```
{//É ano bisexto
```

```
{
```

```
dia_atu=1;
```

```
mes_atu=3;
```

```
{
```

```
dia_atu=1;
```

```
mes_atu=3;
```

```
if (dia_atu>29)
```

```
}
```

```
}
```

```
else {
```

```
if (dia_atu>28)
```

```
}
```

```
}
```

```
break;
```

```
case 3:
```

```
if (dia_atu>31) {
```

```
dia_atu=1;
```

```
mes_atu=4;
```

```
}
```

```
break;
```

```
case 4:
```

```
if (dia_atu>30) {
```

```
dia_atu=1;
```

```
mes_atu=12;
```

```
}
```

```
break;
```

```
case 5:
```

```
if (dia_atu>31) {
```

```
dia_atu=1;
```

```
mes_atu=6;
```

```
}
```

```
break;
```

```
case 6:
```

```
if (dia_atu>30) {
```

```
dia_atu=1;
```

```
mes_atu=12;
```

```
}
```

```
break;
```

```
case 7:
```

```
if (dia_atu>31) {
```

```
dia_atu=1;
```

```
mes_atu=8;
```

```
}
```

```

        break;
    case 8:
        if (dia_atu>31) {
            dia_atu=1;
            mes_atu=9;
        }
        break;
    case 9:
        if (dia_atu>30) {
            dia_atu=1;
            mes_atu=12;
        }
        break;
    case 10:
        if (dia_atu>31) {
            dia_atu=1;
            mes_atu=11;
        }
        break;
    case 11:
        if (dia_atu>30) {
            dia_atu=1;
            mes_atu=12;
        }
        break;
    case 12:
        if (dia_atu>31) {
            dia_atu=1;
            mes_atu=1;
            ano_atu++;
        }
        break;
    }
}

}

if
((seg_atu==seg_fut)&&(min_atu==min_fut)&&(hor_atu==hor_fut)&&
(dia_atu==dia_fut)&&(hor_atu==hor_fut)&&(ano_atu==ano_fut))
    { //Chegou o instante do evento programado
        break;
    }
}
} //while(1);
}

```

```

/* Programa Principal */
void main() {
    /* Inicializa variáveis */
    int k,x;
    int num_toques=3;

    /* Inicializa os bits de controle em todas as portas */
    P0=0;
    P1_1=0; //Buzzer desativado
    P1_7=0; //Apaga led sinalizador de software
    P1_2=0; //LOOP aberto

    P3_6=1; //Decodificador de tons DTMF (MT8870) Habilitado
    P1_5=1; //Gerador de tons DTMF (pino CE do HT9200) Desabilitado
    P1_6=1; //Gerador de tons DTMF (pino S/P do HT9200) no modo paralelo
    ALE=ALE_OFF;
    IE=0; //Desabilita todas as interrupções

    /* Loop infinito, até o instante do evento programado */
    while(1) {
        if (P1_0==0) { //Se recebeu RING

            P1_1=1; //Ativa Buzzer
            espera_nx50ms(20); //Retardo de 1 segundo
            P1_1=0; //Desativa Buzzer

            P1_2=1; //Fecha o LOOP (prepara para recebimento da programação)
            recebe_programacao();
            P1_2=0; //Abre o LOOP

            num_interv_50ms=20; //Inicializa contador de interrupções do timer2
                                //para gerar uma base de
tempo de 1 segundo

            IE=0xA0; //Habilita a interrupção do Timer2
            inic_t2_50ms(); //Inicializa o Timer2

            monitora_evento(); //Passa a controlar o tempo para a execução do
evento

            IE=0x00; //Desabilita todas as interrupções

            P1_2=1; //Fecha o LOOP para receber o tom de discar

            buzz(4);

            espera_nx50ms(40); //Espera (2 segundos) pelo tom de
discar;

```

```

P1_7=1;    //Sinaliza LED

if (P3_7==0) {
    num_toques=1;
}

x=num_toques;
do { /* Disca para o número programado */
    for (k=0; k<8; k++) {
        P0=NUMDISCAR[k];
        P2=NUMDISCAR[k]&0x0F;
        P1_5=0;    //Habilita o gerador de tons DTMF
        espera_nx50ms(4); //Espera 200 milisegundos
        P1_5=1;    //Desabilita o gerador de tons DTMF
        espera_nx50ms(4); //Espera 200 milisegundos
    }
    espera_nx50ms(60);    //Espera 3 segundos
}while(--x>0);

espera_nx50ms(80);    //Espera 4 segundos
buzz(10);    //Sinaliza na saída do programa
espera_nx50ms(40);
P1_7=0;    //Apaga o LED da sinalização
P1_2=0;    //Abre o LOOP
break;    //Sai do programa
}
}
}

```

Anexo 2 – Kit de Microcontrolador

KIT DE MICROCONTROLADOR

MANUAL DE INSTRUÇÕES

NEUROTREND - Technology Solutions Ltda.

www.microcontrolador.com.br

VERSÃO 1.0 – 2002

Principais Características:

-Permite comunicação com o PC via serial e também via Paralela do PC, neste caso para gravação do Microcontrolador AT89S8252;

-Gravação ISP – In System Programming, do AT89S8252;

-Chave de LOAD / RUN com LEDs para facilitar o processo de “gravação” e “rodagem” do programa desenvolvido, sem precisar tirar cabos de comunicação ISP ou o SERIAL;

-8 Saídas de I/O Mapeado (para usar,gravar linhas de programação conforme descrito em “Mapas de Endereços” neste manual).

-8K de Flash

-2K EEPROM

-4 ports disponíveis com Conector;

-Comunicação Serial RS232 inclusa;

-Cabos Disponíveis:

Extensão dos Ports (2 unidades),

Comunicação Serial(1 unidade),

Gravação ISP(1 unidade).

-Alimentação Necessária: 9V DC , 500 mA (não inclusa)

<http://www.microcontrolador.com.br>
suporte@microcontrolador.com.br

ADVERTÊNCIA

As informações e o material contidos neste manual são fornecidos sem nenhuma garantia, quer explícita, ou implícita, de que o uso de tal informação conduzirá sempre ao resultado desejado. Todos os envolvidos na confecção deste material não podem ser responsabilizados por qualquer tipo de reivindicação atribuída a erros, omissões ou quaisquer outras imprecisões na informação ou material fornecidos neste manual, e em nenhuma hipótese podem ser incriminados direta ou indiretamente por qualquer dano, perda, lucro cessante, etc, devido ao uso destas

informações.

Para utilização do material com finalidades didáticas, de ensaios, teste e/ou programação, o usuário deve ter mínimos conhecimentos técnicos sobre eletrônica, bem como conhecer os princípios de microprocessadores, microcontroladores, compiladores e simuladores, além de ter algum tipo de experiência com montagem de placas de circuito impresso, reconhecimento de componentes eletrônicos, lógica digital, soldagem e familiaridade com computadores do tipo PC e sistema operacional DOS/Windows. Sem estes pré-requisitos é muito provável que o usuário tenha diversas dificuldades durante os testes de funcionamento do Kit.

APRESENTAÇÃO

Este sistema tem o objetivo de facilitar o aprendizado do **Microcontrolador da família 8051 da ATMEL** intitulada **AT89S8252**, e com este KIT agilizar o desenvolvimento de experiências e projetos em termos didático e profissional.

INTRODUÇÃO

O crescente interesse por essa família de microcontroladores deve-se ao fato de serem baratos, poderosos, facilmente encontrados e com uma grande variedade de microcontroladores que contém características especiais como conversores A/D, saídas PWM, etc.; e também por um mercado potencialmente crescente.

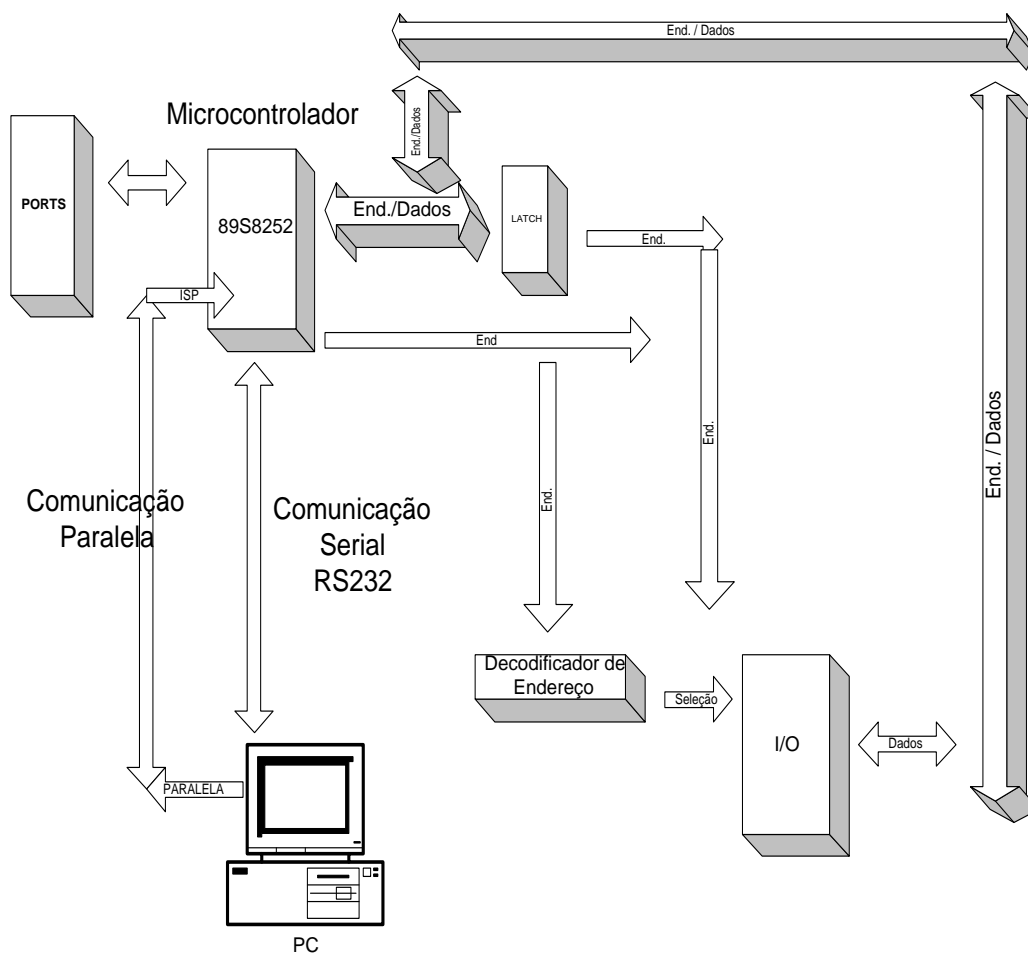
Apesar de ser um microcontrolador de 8 bits, pode-se desenvolver sistemas dedicados poderosos a um custo muito reduzido.

O sistema apresentado é extremamente flexível, pois todos os **ports** estão disponíveis ao usuário o que facilita estender as características da placa de acordo com a necessidade.

O sistema apresentado permite comunicação por técnica conhecida como I/O Mapeado, podendo estender os ports deste micro até 8 saídas endereçadas com endereço fixo (8000h até 8007h), usando para isso os **ports** P0 e P2 parcialmente, e os pinos Rd e WR do Port P3.

Permite também executar o programa desenvolvido em uma memória Flash do AT89s8252.

O diagrama a seguir demonstra os principais componentes e fluxo de dados:

DIAGRAMA DE BLOCOS

ESQUEMA DA PLACA

PARA VISUALIZAÇÃO, VEJA ARQUIVO SCH89S8252V1_02.PDF

DESCRIÇÃO DA CPU

Esta placa contém um microcontrolador de 8 bits (AT89S8252) que possui internamente 5 interrupções mascaráveis, 2 timers de 16 bits programáveis e um canal serial, que já tem dentro dele uma Flash de 8K e EEPROM de 2K Bytes.

Possui também oito endereços reservados para operações de I/O Mapeado.

Além disso pode ser comandado por uma simples chave LOAD/RUN com LEDs indicadores, para se gravar e rodar os programas.

Portanto para transmitir o programa do PC para o Kit é necessário que a chave esteja na posição LOAD, que através do Conector ISP que se conecta na paralela do PC até a entrada ISP do Kit, se possa gravar a Flash do Atmel. Após a gravação, posicionar a chave HH1 para **RUN** para rodar o programa gravado, e não precisa desconectar o cabo de ISP, facilitando e agilizando o processo de desenvolvimento e/ou aprendizado.

MAPEAMENTO DE ENDEREÇOS EXTERNOS

Endereço	Função
0000H – 7FFFH	Região Não utilizada
8000H - 8007H	I/O mapeado
8008H - EFFFH	Região Não utilizada
FFF0H - FFF3H	I/O externa
FFF4H - FFFFH	Região Não utilizada

OBS: Quando tiver que usar o I/O mapeado, inserir no início de cada programa desenvolvido os códigos abaixo, a fim do chip entender que deve usar P0 no modo multiplexado (endereços + dados em P0) :

```

ORG 0000H
    mov A, 8Eh                ;carrega registrador "8E" h em
A.
    ORL A, #00000001b        ;força bit 0 deste Registrador
ser "1".
    mov 8Eh, A                ;devolve para Reg"8E" h seu
    valor já alterado o
                                ;Bit 0 para ser 1.
                                ;
                                ;continuar o programa a ser
                                desenvolvido....

```

ATENÇÃO:

Cada vez que se usa o canal serial do Kit (RS232), deve-se antes posicionar a chave HH2 para "fora" da placa. Isto faz com que os pinos P3.0 e P3.1 do conector "Port P3" fiquem desconectados deste conector, para evitar acionamentos indevidos que possa ocorrer através de um cabo que possa estar ligado no conector do Port P3 durante os instantes de transmissão e recepção da serial.

É saudável, também, que na transmissão de gravação do arquivo, o conector do Port P1 no Kit esteja livre (sem estar ligado a qualquer circuito externo ao Kit). O motivo disto é que, como já citado, o Kit utiliza parte do port P1 para transferência do arquivo do PC ao Atmel. Se existir algum circuito externo ligado em P1, no momento de transferência do programa do PC ao Kit, este circuito externo pode alterar a transmissão resultando em arquivo errado, que resultará em funcionamento não adequado.

ROTINA PARA GRAVAR O PROGRAMA NO ATMEL

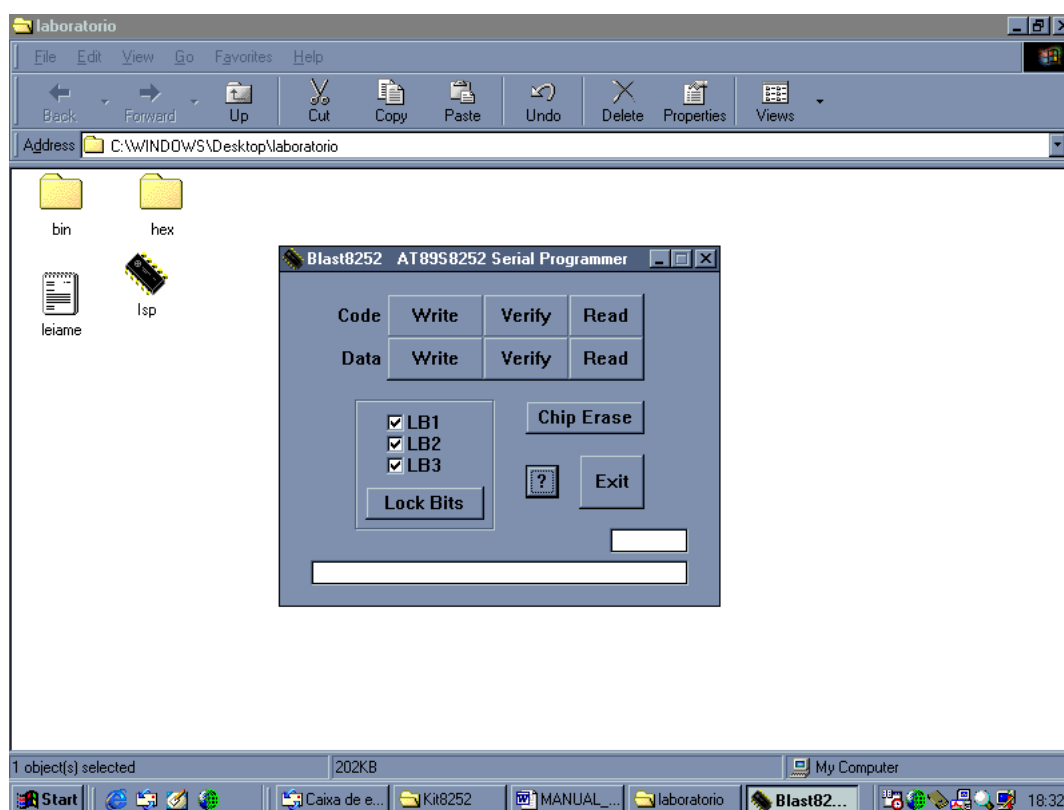
Primeiramente, obter um compilador de 8051 e simulador e o software de gravação ISP da Flash via comunicação paralela de um PC (vide site da Neurotrend www.microcontrolador.com.br, em downloads, na pasta kit Atmel AT89S8252)

Localizar a porta paralela do seu PC que será utilizada para comunicação

PC ↔ KIT (conector ISP) e ligar o Cabo ISP (este cabo vem com o Kit e tem conector de 5 vias em uma ponta e um DB25 na outra, enquanto que o outro cabo que vem com o kit é o cabo SERIAL, que é quase igual, mas vem com conector de 3 vias como será visto posteriormente) entre a saída paralela do PC e o conector intitulado “ISP” ou “CON02” de 5 pinos, do Kit.

Com a placa ligada e conectada no PC e a chave “LOAD-RUN” ligada em Load, e o Kit ligado na alimentação, você deve usar o programa gravador intitulado “ISP” e o programa exemplo “pisca1_3.bin” para gravar o Atmel. O Kit já vem de fábrica testado e com este programa gravado, logo você pode verificar se o Kit está em ordem ligando-o e observando com multímetro ou osciloscópio que os pinos de P1 e P3 devem estar piscando com intervalo aproximado de 1 segundo.

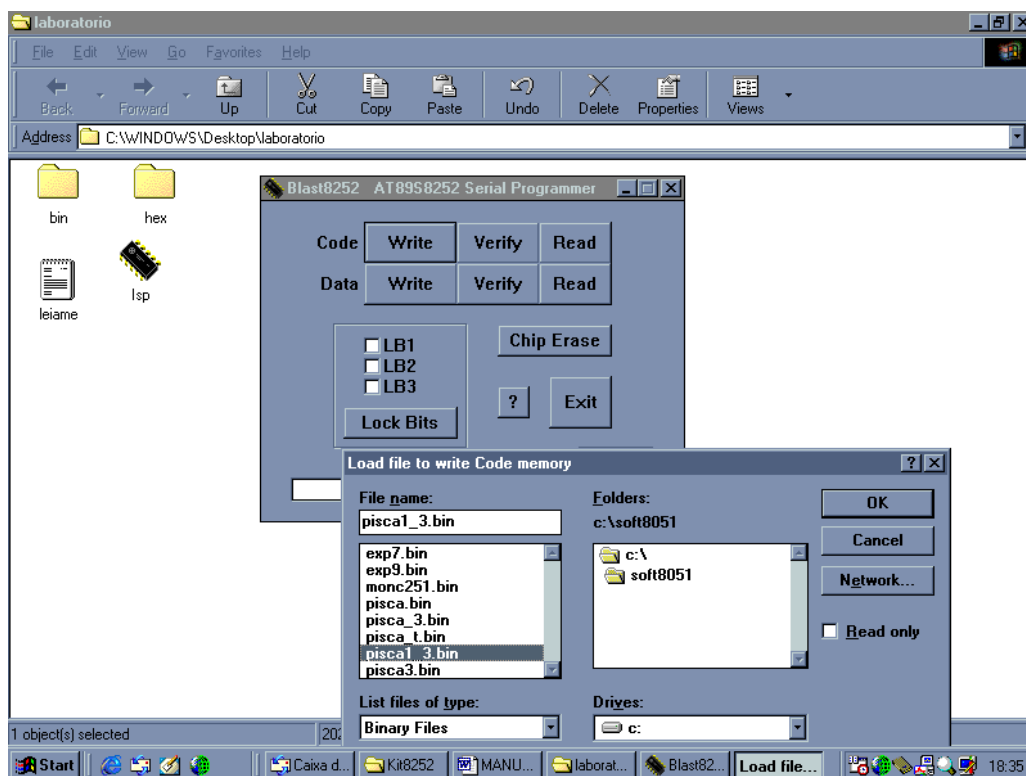
O programa ISP tem as seguintes telas:



TELA 1: Tela de abertura do executável intitulado “ISP”

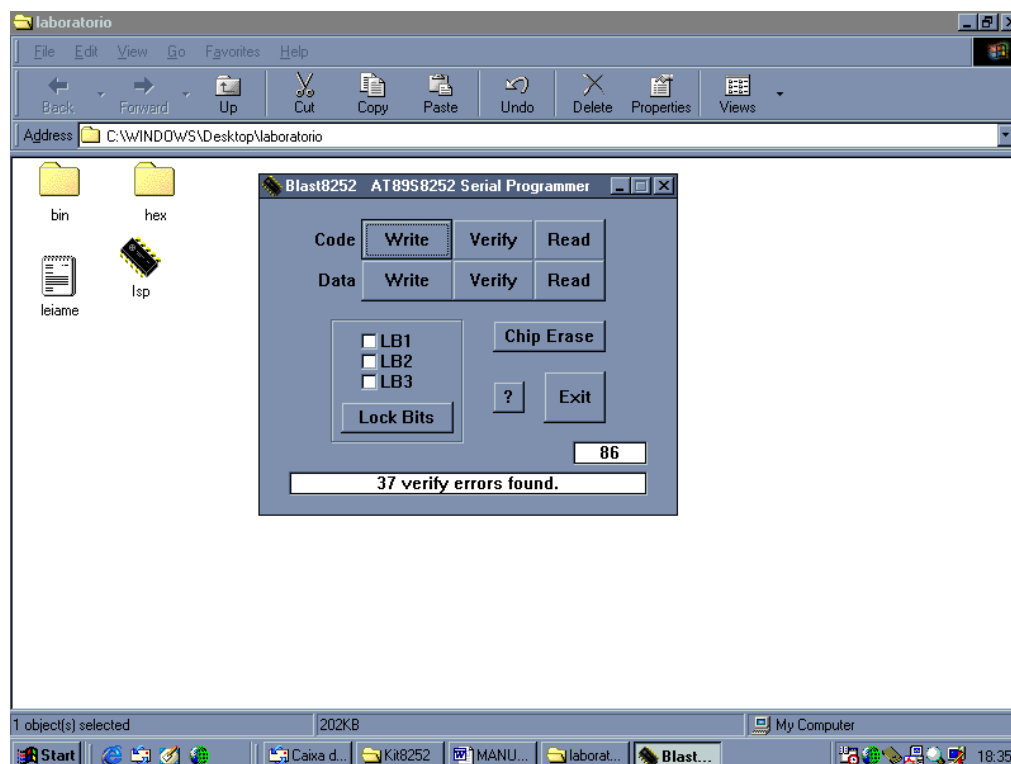
A primeira coisa a fazer é clicar nos “Lock Bits” e apagá-los. Para entender

sobre os “Lock Bits” (proteção com segredo do programa gravado, leia um Manual do AT89S8252, pois este assunto não será tratado aqui).



TELA 2: Clicar no botão “Code Write” e aparecerá tela para se procurar arquivos já compilados (só extensão “.bin”). No site temos dois arquivos de teste, os “pisca_3.bin” e “pisca1_3.bin” que respectivamente fazem os ports P1, e no outro arquivo os P1 e P3, piscarem em freqüência aproximada de 1 segundo. Escolha seu programa já desenvolvido e compilado e dê o OK. Não esquecer de apagar, antes de gravar, os “Lock Bits”.

ATENÇÃO: O KIT JÁ VEM DE FÁBRICA COM O ARQUIVO “PISCA1 3.BIN” GRAVADO. ASSIM, A PRIMEIRA COISA A SE FAZER É TESTAR O KIT, JUNTO COM SUA FONTE DE ALIMENTAÇÃO, PARA VER SE ELE ESTÁ EM ORDEM E SÓ DEPOIS USAR O PROGRAMA “ISP” PARA REGRAVAR O ATMEL QUE ACOMPANHA O KIT.



TELA 3: Apertar o botão “Code Write” e esperar ele transmitir o arquivo e dar click no botão “OK” . Se der erro na transmissão, faça os seguintes passos:

- 1-Apague o chip antes de usa-lo para gravar, com o botão “Chip Erase”.
- 2-Tente escolher de novo o arquivo desejado e gravar de novo no chip, após item 1 realizado.
- 3- Se apresentar erros, tente agora, de novo, gravá-lo sem apagar, pois pode ter deficiências de gravação que se supera gravando mais que uma vez o mesmo arquivo.
- 4- Se continuar dando erros, verifique se a chave “Load-Run” está na posição “Load” e se o “Led” correspondente está aceso.
- 5- Se nada funcionar, verifique as condições da fonte DC utilizada no Kit, e se sua saída paralela do Micro está em ordem (teste-a usando sua impressora nela, por exemplo).
- 6- Se a repetição cuidadosa destes itens não levarem ao sucesso, contate o apoio técnico do site www.microcontrolador.com.br via email para seu suporte técnico.

A área de gravação de EPROM (Botão “Data Write” na tela do ISP): é como gravar a Flash, já explicada acima, porém estes passos não serão detalhados aqui, pois o usuário deverá antes estudar o manual do chip AT89S8252, para saber onde e como criar os arquivos para esta gravação, e também como utilizar estes Bytes gravados no próprio programa que foi desenvolvido.

Como criar um Programa a ser Compilado:

Deve-se digitar o programa em qualquer editor de texto que grave o arquivo em formato “puro ASCII”, por exemplo, o “EDIT” do DOS ou o Bloco de Notas (Note Pad) do Windows. O programa “Word Pad” não serve, pois não é ASCII puro. Salvar o arquivo com nome de até 8 letras, com extensão “**.ASM**”, que é a extensão do arquivo que se utiliza na compilação.

Em seguida deve-se compilar e linkar o programa utilizando o comando **RODA51** que é um arquivo de lote criado para realizar a compilação, linkagem e transformar seu programa em um arquivo compilado com extensão “**.BIN**”.

Esse arquivo será transmitido do PC para a placa utilizando-se o programa em Windows intitulado “**ISP.EXE**”.

Para executar estas operações deve-se seguir os seguintes passos:

1- Editar seu programa, por exemplo, criando e salvando o arquivo **PISCA1_3.ASM** (Este exemplo já está pronto no site www.microcontrolador.com.br em “downloads” na área do “Kit Atmel AT89S8252”). O KIT JÁ VEM DE FABRICA GRAVADO COM ESTE PROGRAMA DE TESTE.

2- Digitar na linha de comando do “Prompt” do DOS, e direcionar para a mesma área de diretório que você gravou o programa acima e salvou os

arquivos que pegou no nosso site (entrar no “DOS” para isso). O exemplo ilustrativo abaixo supõe que você salvou todos os arquivos no diretório C:\8051.

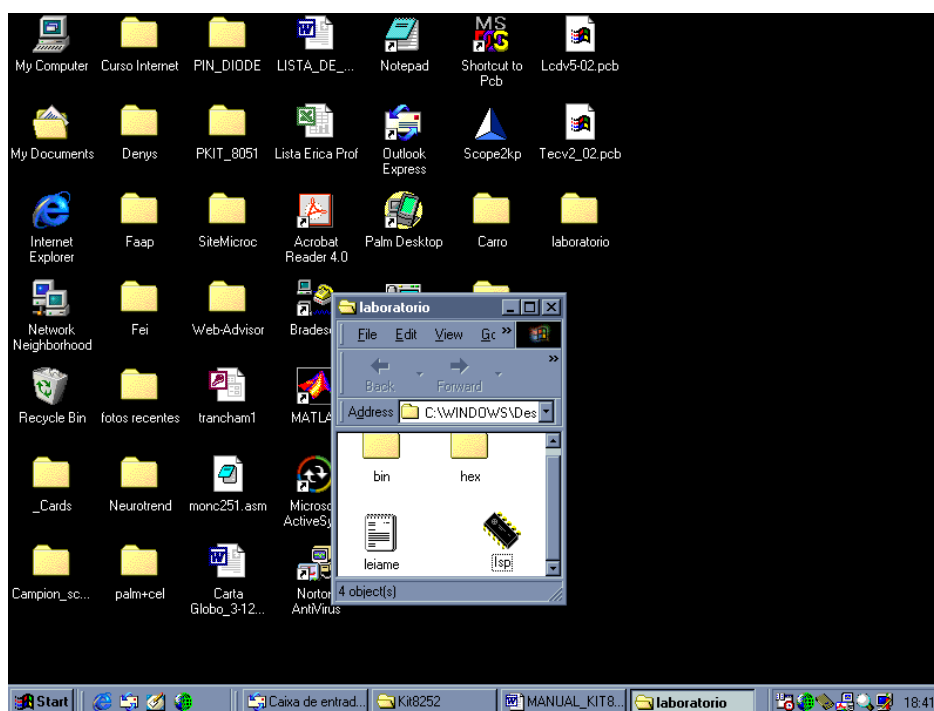
Para Compilar e descobrir erros:

C:\8051>RODA51 PISCA1_3 (NÃO UTILIZAR EXTENSÃO)

3- Se houver erros, procurá-los no arquivo correspondente com extensão “.LST” (em nosso exemplo seria o arquivo *PISCA1_3.LST*) que o programa criou para te mostrar os erros encontrados. Use o comando abaixo para abrir o arquivo:

C:\8051>EDIT PISCA1_3.LST

4- Se não houver erros, ele criará o arquivo “PISCA1_3.BIN” que deverá ser enviado para o Kit através dos seguintes comandos: (Abrir o windows explorer, localizar o arquivo “isp.exe” ou seu ícone e acioná-lo) :



TELA inicial: Ir agora para o capítulo intitulado “**ROTINA PARA GRAVAR O PROGRAMA**”, **DESTE MANUAL.**

Comunicação Serial RS232 do Kit com um PC

Pode-se realizar a comunicação do Kit com qualquer serial que fale RS232, ou até com outro kit. Aqui vamos explorar o caso da comunicação do kit com um PC, que é o caso mais utilizado tipicamente.

Cabo de Conexão para uso da Serial do kit:

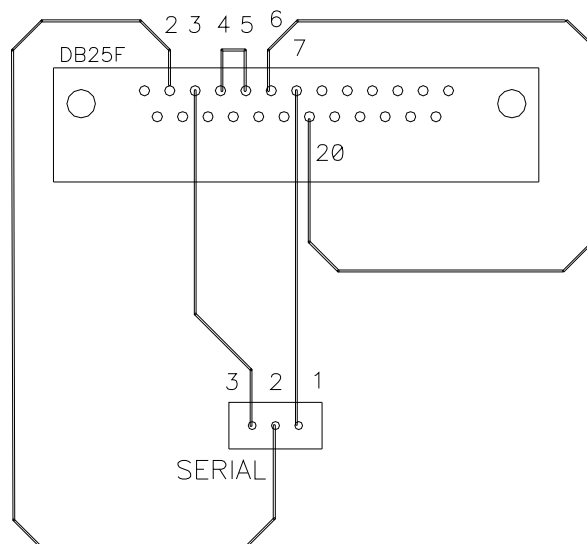
Você recebe o Kit com um cabo pronto para a COM2.

Para conexão na COM2 o cabo vem ligado conforme a seguir:

- Pino 2 da COM2 é ligado no pino 2 da placa;
- Pino 3 da COM2 é ligado no pino 1 da placa;
- Os terras são ligados para que não exista uma diferença entre terras, o que poderia danificar os equipamentos. (pino 7 da COM2 deve ser ligado ao pino 3 da placa)
- Observe também os curto-circuitos que foram feitos entre os pinos 6-20 e 4-5 no conector COM2 para permitir o correto funcionamento da comunicação KIT ↔ PC.

Esquema da conexão serial para conector de 25 pinos (COM2):

Onde pino 2 é TX , pino 3 é RX e pino 7 é GND. (PC)



Onde pino 3 é TX, pino 2 é RX e pino 1 é GND. (Na placa)

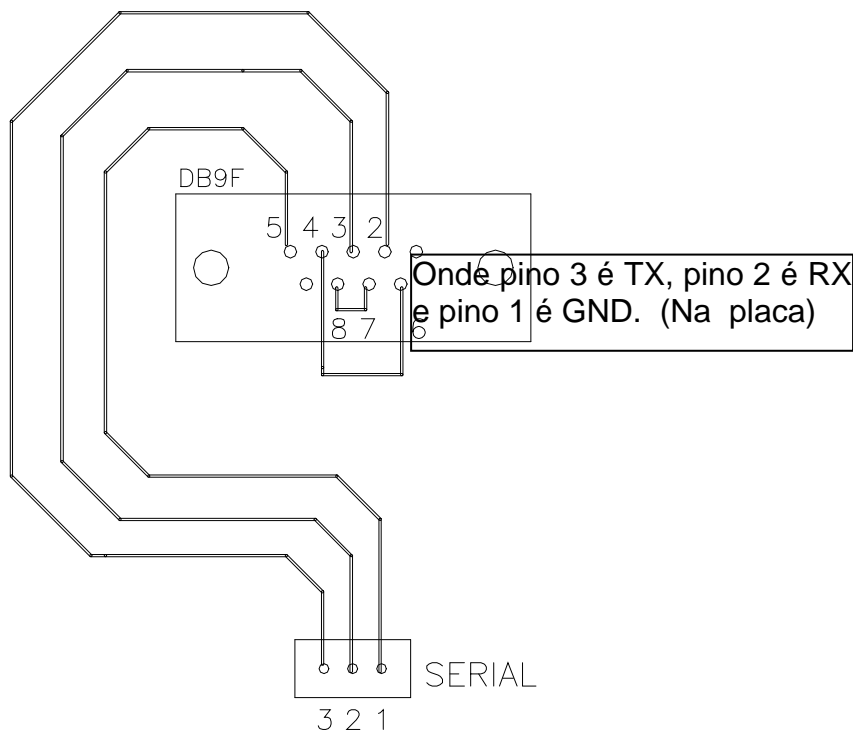
PC	KIT ATMEL
TX (pino 2 na COM2:)	RX da placa (pino 2)
RX (pino 3 na COM2:)	TX da placa (pino 3)
GND (pino 7 na COM2:)	GND da placa (pino 1)

Para conexão na COM1

Use um adaptador de 25 para 9 pinos que se acopla ao cabo recebido, ou faça um novo cabo conforme a seguir:

- Pino 3 da COM1 deve ser ligado no pino 2 da placa;
- Pino 2 da COM1 deve ser ligado no pino 1 da placa;
- Os pinos de terra devem ser ligados para que não exista uma diferença entre terras, o que poderia danificar os equipamentos. (pino 5 da COM1 deve ser ligado ao pino 3 da placa)
- Observar também os curto-circuitos entre os pinos 7-8 e 4-6 no conector COM1 a fim de permitir a correta comunicação entre KIT ↔ PC.

Esquema da conexão serial para conector de 9 pinos (COM1):
Onde pino 3 é TX, pino 2 é RX e pino 5 é GND. (No PC)



PC	KIT ATMEL
TX (pino 3 na COM1:)	RX da placa (pino 2)
RX (pino 2 na COM1:)	TX da placa (pino 3)
GND (pino 5 na COM1:)	GND da placa (pino 1)

Conecte o cabo serial na placa do Kit e na porta serial do PC.

Após ter gravado seu programa no Kit, que aqui se supõe que seu programa utilizará a comunicação serial do mesmo, conecte o cabo que recebeu com o Kit intitulado “Cabo Serial” (ele é um cabo com conector DB25 em uma ponta (para ligar na COM2 do PC) com a outra ponta com

terminal com 3 pinos fêmea, cujos irão se conectar no Kit, no conector nomeado na placa como “SERIAL” ou “CON01”). Você irá deslocar a chave HH1 **da posição “LOAD” (considera-se que já acabou a gravação) para a posição RUN (ou para “dentro” da placa)** mas antes e também mude a chave HH2 para a posição “HH2” escrita na PCI (ou para “fora” da placa).

Após então ter acertado a posição de HH2, desloque em seguida a chave HH1 para a posição **“RUN” (contrário da posição “LOAD”)** para que o microcontrolador comece a executar seu programa de comunicação entre o Kit e o PC, aqui neste exemplo.

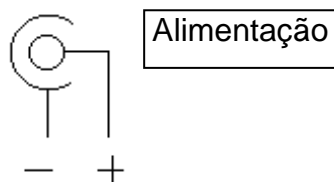
Para carregar um novo programa desloque novamente a chave HH1 para a posição **LOAD** e grave seu novo programa para a placa utilizando o programa **ISP.EXE**. A chave HH2 não precisa mudar mais de posição, a não ser que o programa que será executado venha a utilizar os pinos de comunicação serial (P3.0 e P3.1). Neste caso, cada vez que se muda a chave HH1 também se deve mudar a chave HH2, respectivamente, pois assim ela liberará os pinos P3.0 e P3.1 que estão ligados à comunicação serial RS232 do PC.

Alimentação:

Recomenda-se a utilização de uma fonte de 9V DC com $I > 500\text{mA}$ **(ou equivalente confiável, sendo pino interno positivo que se usa é “pino fino” neste conector , de 2,1 mm. Tem fontes que vem com pino grosso, de 2,5 mm, e dará mau contato ou dificuldades na ligação, fazendo o programa cair toda hora).** A figura a seguir demonstra como deve ser o conector: Observe que em algumas fontes o conector de saída tem escrito os sinais de “+” e “-“. A polaridade do mesmo pode ser mudada desconectando o plug do próprio cabo e invertendo-o.

UTILIZA-SE PINO INTERNO “FINO” (2,1mm) NO CONECTOR DE

ALIMENTAÇÃO. PORTANTO, VOCE DEVE TER FONTE COM CONECTOR J4 COM PINO FINO. A OUTRA DIMENSÃO EXISTENTE NO MERCADO É DE 2,5mm, QUE É FACILMENTE PASSADO, A OLHO NÚ, COMO PINO FINO E NÃO ENTRA NO CONECTOR.



Programa exemplo/teste do KIT8051 :

Vide arquivos “pisca1_3.asm” e “pisca_3.asm” no site www.microcontrolador.com.br na área de download, kit atmel AT89S8252.

ESQUEMA :

VIDE ARQUIVO “PDF” NO SITE www.microcontrolador.com.br, na mesma área de arquivos deste kit.

Anexo 3 – Datasheet receptor DTMF MT8870